aws

Guida per sviluppatori di database

Amazon Redshift

Amazon Redshift: Guida per sviluppatori di database

Table of Contents

Introduzione

Benvenuti nella Guida per gli sviluppatori di database di Amazon Redshift. Amazon Redshift è un servizio di data warehouse nel cloud in scala petabyte interamente gestito. Amazon Redshift serverless consente di accedere e analizzare i dati senza le consuete configurazioni di un data warehouse con provisioning. Viene eseguito automaticamente il provisioning delle risorse e la capacità del data warehouse viene dimensionata in modo intelligente per fornire prestazioni rapide per carichi di lavoro maggiormente impegnativi e imprevedibili. Quando il data warehouse è inattivo non vengono addebitati costi, si paga solo l'utilizzo. Indipendentemente dalle dimensioni del set di dati, è possibile caricare i dati e iniziare subito a eseguire query nell'editor di query Amazon Redshift v2 o nello strumento di business intelligence (BI) preferito. Goditi il miglior rapporto prezzo/prestazioni e le familiari funzionalità SQL in un easy-to-use ambiente senza amministrazione.

In questa guida viene descritto come utilizzare Amazon Redshift per creare e gestire un data warehouse. Se utilizzi database in quanto progettista, sviluppatore software o amministratore, questa guida ti offre le informazioni necessarie per progettare, creare, sottoporre a query e gestire il tuo data warehouse.

Argomenti

Prerequisiti

Prima di utilizzare questa guida, dovresti leggere Amazon Redshift serverless, che illustra come completare le seguenti attività.

- Creazione di un data warehouse con Amazon Redshift serverless.
- Caricamento di dati di esempio con l'editor di query Amazon Redshift v2
- Caricamento di dati da Amazon S3.

Devi inoltre sapere come utilizzare il client SQL e avere una conoscenza di base del linguaggio SQL.

Sei uno sviluppatore di database?

Se si utilizza Amazon Redshift per la prima volta, consigliamo di leggere Amazon Redshift serverless per infirmazioni su come iniziare.

Se sei un utente, un progettista, uno sviluppatore o un amministratore di database, la tabella seguente ti aiuterà a trovare quello che cerchi.

Se vuoi...	Consigliamo…
Scopri di più sull'architettura interna del data warehouse di Amazon Redshift.	La Panoramica del sistema e dell'architettura fornisce una panoramica di alto livello dell'architettura interna di Amazon Redshift. Se si desidera una panoramica più ampia del servizio Web di Amazon Redshift, consultare la pagina dei dettagli del prodotto Amazon Redshift.
Creare database, tabelle, utenti e altri oggetti di database.	Common Database Tasks è una rapida introduzione alle basi dello sviluppo di SQL. SQL Amazon Redshift contiene la sintassi e gli esempi di funzioni e comandi SQL di Amazon Redshift nonché altri elementi SQL. Best practice di Amazon Redshift per la progettazione di tabelle fornisce un riepilogo delle nostre raccomandazioni quanto a scelta di chiavi di ordinamento, chiavi di distribuzione e codifiche di compressione.
Apprendere a progettare tabelle per ottenere prestazioni ottimali.	Utilizzo dell'ottimizzazione automatica delle tabelle descrive in modo dettagliato le considerazioni per applicare la compressione ai dati nelle colonne di tabella e per scegliere chiavi di ordinamento e di distribuzione.
Caricare dati.	Nella sezione Caricamento dei dati sono descritte le procedure di caricamento di set di dati di grandi dimensioni da tabelle Amazon DynamoDB o file flat archiviati in bucket Amazon S3. Best practice di Amazon Redshift per il caricamento di dati fornisce suggerimenti per caricare dati in modo rapido ed efficace.

Se vuoi...	Consigliamo…
Gestire utenti, gruppi e la sicurezza dei database.	Gestione della sicurezza del database copre gli argomenti sulla sicurezza dei database.
Monitorare e ottimizzare le prestazioni del sistema.	Riferimento di tabelle e viste di sistema descrive in dettaglio le tabelle e le viste di sistema che puoi sottoporre a query per lo stato del database e per monitorare query e processi. Consulta anche la Guida alla gestione di Amazon Redshift per scoprire come utilizzare la AWS Management Console per verificare lo stato di integrità del sistema, monitorare i parametri, eseguire il backup e ripristinare i cluster.
Analizzare e notificare informazioni da set di dati voluminosi.	Molti fornitori di software noti certificano Amazon Redshift con le loro offerte per consentire l'utilizzo degli strumenti attualmente utilizzati. Per ulteriori informazioni, consultare la pagina dei partner di Amazon Redshift. Documentazione di riferimento a SQL contiene tutti i dettagli sulle espressioni, i comandi e le funzioni SQL che Amazon Redshift supporta.
Interazione con le risorse e le tabelle di Amazon Redshift.	Consulta Amazon Redshift Serverless API guide (Guida alle API di Amazon Redshift Serverless, Amazon Redshift API guide (Guida alle API di Amazon Redshift) e Amazon Redshift Data API guide (Guida alle API dati di Amazon Redshift) per ulteriori informazioni su come è possibile interagire in modo programmatico con le risorse ed eseguire operazioni.
Segui un tutorial per acquisire maggiore dimestichezza con Amazon Redshift.	Segui un tutorial nella sezione dei tutorial per Amazon Redshift per ulteriori informazioni sulle funzionalità di Amazon Redshift.

Panoramica del sistema e dell'architettura

Un data warehouse di Amazon Redshift è un sistema di query e gestione di database relazionali di livello aziendale.

Amazon Redshift supporta connessioni client con molti tipi di applicazioni, tra cui strumenti di business intelligence, creazione di report, gestione dei dati e analisi.

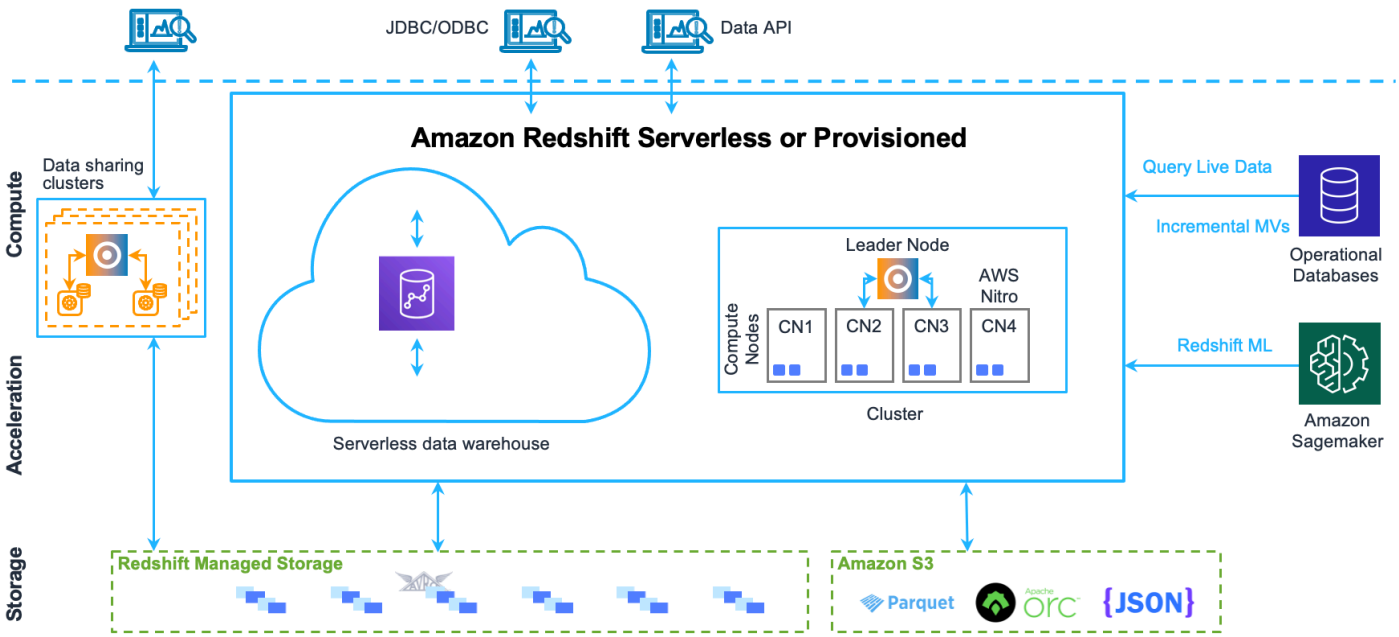
Attraverso l'esecuzione di query di analisi, puoi recuperare, confrontare e valutare grandi quantità di dati in operazioni in più fasi per produrre un risultato finale.

Amazon Redshift garantisce storage efficiente e prestazioni di esecuzione di query ottimali tramite una combinazione di Massively Parallel Processing, storage dei dati a colonne e schemi di codifica di compressione dei dati mirati molto efficaci. Questa sezione presenta un'introduzione all'architettura del sistema Amazon Redshift.

Argomenti

Architettura del sistema di data warehouse

Questa sezione presenta gli elementi dell'architettura di data warehouse di Amazon Redshift, mostrata nella figura seguente.



Applicazioni client

Amazon Redshift si integra con diversi strumenti di caricamento dei dati ed ETL (estrazione, trasformazione e caricamento) nonché creazione di report, data mining e analisi di Business Intelligence (BI). Amazon Redshift è basato sullo standard aperto PostgreSQL, pertanto sarà possibile utilizzare la maggior parte delle applicazioni client SQL esistenti apportando solo modifiche minime. Per informazioni su alcune importanti differenze tra SQL e PostgreSQL in Amazon Redshift, consultare Amazon Redshift e PostgreSQL.

Cluster

Il componente centrale dell'infrastruttura di un data warehouse di Amazon Redshift è un cluster.

Un cluster è costituito da uno o più nodi di calcolo. Se viene effettuato il provisioning di un cluster con due o più nodi di calcolo, un ulteriore nodo principale coordina i nodi di calcolo e gestisce la comunicazione esterna. L'applicazione client interagisce direttamente solo con il nodo principale. I nodi di calcolo sono trasparenti alle applicazioni esterne.

Nodo principale

Il nodo principale gestisce le comunicazioni con i programmi client e tutta la comunicazione con i nodi di calcolo. Questo nodo analizza e sviluppa piani di esecuzione per svolgere operazioni di database, in particolare la serie di operazioni necessarie per ottenere risultati per query complesse. In base al piano di esecuzione, il nodo principale compila il codice, distribuisce il codice compilato nei nodi di calcolo e assegna una parte dei dati a ogni nodo di calcolo.

Il nodo principale distribuisce istruzioni SQL nei nodi di calcolo solo quando una query fa riferimento a tabelle archiviate nei nodi di calcolo. Tutte le altre query vengono eseguite esclusivamente nel nodo principale. Amazon Redshift è progettato per implementare determinate funzioni SQL solo nel nodo principale. Una query che usa una di queste funzioni restituirà un errore se fa riferimento a tabelle che si trovano nei nodi di calcolo. Per ulteriori informazioni, consultare Funzioni SQL supportate sul nodo principale.

Nodi di calcolo

Il nodo principale compila il codice per singoli elementi del piano di esecuzione e assegna il codice a singoli nodi di calcolo. I nodi di calcolo eseguono il codice compilato e restituiscono risultati intermedi al nodo principale per l'aggregazione finale.

Ogni nodo di calcolo ha CPU e memoria dedicati, determinati dal tipo di nodo. Con il crescere del carico di lavoro, è possibile incrementare la capacità di calcolo di un cluster aumentando il numero di nodi, aggiornando il tipo di nodi o con entrambe queste operazioni.

Amazon Redshift fornisce diversi tipi di nodi per soddisfare le esigenze di calcolo degli utenti. Per informazioni dettagliate su ciascun tipo di nodo, consulta Cluster di Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Redshift Managed Storage

Per l'archiviazione dei dati di data warehouse viene utilizzato Redshift Managed Storage (RMS) come livello di archiviazione separato. RMS offre la possibilità di dimensionare l'archiviazione fino ai petabyte utilizzando l'archiviazione di Amazon S3. RMS consente di dimensionare e pagare per il calcolo e l'archiviazione in modo indipendente, pertanto è possibile dimensionare il cluster solo in base alle esigenze di calcolo. Utilizza automaticamente l'archiviazione locale basata su SSD ad alte prestazioni come cache di livello 1. Sfrutta inoltre ottimizzazioni quali temperatura di blocco dei dati, età di blocco dei dati e modelli di carichi di lavoro per offrire prestazioni elevate, oltre a dimensionare l'archiviazione automaticamente in Amazon S3 quando necessario senza richiedere alcuna operazione da parte dell'utente.

Sezioni dei nodi

Ogni nodo di calcolo è partizionato in sezioni. A ogni sezione viene allocata una parte della memoria e dello spazio su disco del nodo, in cui elabora una parte del carico di lavoro assegnato al nodo. Il nodo principale gestisce la distribuzione dei dati nelle sezioni e spartisce tra le sezioni il carico di lavoro per tutte le query o altre operazioni di database. Le sezioni operano quindi in parallelo per completare l'operazione.

Il numero di sezioni per ogni nodo è determinato dalle dimensioni dei nodi del cluster. Per ulteriori informazioni sul numero di sezioni per ogni dimensione di nodo, consulta la pagina relativa alle informazioni su cluster e nodi nella Guida alla gestione di Amazon Redshift.

Quando crei una tabella, puoi facoltativamente specificare una colonna come chiave di distribuzione. Quando carichi dati nella tabella, le righe vengono distribuite nelle sezioni dei nodi in base alla chiave di distribuzione definita per una tabella. La scelta di una chiave di distribuzione appropriata permette ad Amazon Redshift di usare l'elaborazione parallela per caricare dati ed eseguire query in modo efficiente. Per informazioni sulla scelta di una chiave di distribuzione, consultare Scelta del migliore stile di distribuzione.

Rete interna

Amazon Redshift sfrutta connessioni a larghezza di banda elevata, prossimità ravvicinata e protocolli di comunicazione personalizzati per fornire comunicazione di rete privata ad altissima velocità tra il nodo principale e i nodi di calcolo. I nodi di calcolo vengono eseguiti in una rete isolata separata, cui le applicazioni client non accedono mai direttamente.

Database

Un cluster contiene uno o più database. I dati utente vengono archiviati nei nodi di calcolo. Il client SQL comunica con il nodo principale, che a sua volta coordina l'esecuzione di query con i nodi di calcolo.

Amazon Redshift è un sistema di gestione di database relazionali (RDBMS, Relational Database Management System) e di conseguenza è compatibile con altre applicazioni RDBMS. Benché offra le stesse funzionalità delle tipiche applicazioni RDBMS, tra cui funzioni di elaborazione di transazioni online (OLTP) come l'inserimento e l'eliminazione di dati, Amazon Redshift è ottimizzato per analisi e creazione di report ad alte prestazioni di set di dati di dimensioni molto grandi.

Amazon Redshift è basato su PostgreSQL. Amazon Redshift e PostgreSQL si differenziano per un certo numero di aspetti molto importanti, di cui devi tenere conto mentre progetti e sviluppi le tue applicazioni di data warehouse. Per informazioni sulle differenze tra Amazon Redshift SQL e PostgreSQL, consultare Amazon Redshift e PostgreSQL.

Prestazioni

Amazon Redshift garantisce un'esecuzione di query estremamente rapida tramite l'uso di queste caratteristiche delle prestazioni.

Argomenti

- Architetture MPP (Massively Parallel Processing)
- Storage dei dati a colonne
- Compressione dei dati
- Ottimizzatore di query
- Caching dei risultati
- Codice compilato

Architetture MPP (Massively Parallel Processing)

MPP (Massively Parallel Processing) permette la rapida esecuzione delle query più complesse che operano su grandi quantità di dati. Più nodi di calcolo gestiscono l'elaborazione di tutte le query, producendo l'aggregazione dei risultati finali, con ogni core di ciascun nodo che esegue gli stessi segmenti compilati su parti dell'intero set di dati.

Amazon Redshift distribuisce le righe di una tabella nei nodi di calcolo in modo che i dati possano essere elaborati in parallelo. Selezionando una chiave di distribuzione appropriata per ogni tabella, puoi ottimizzare la distribuzione dei dati per bilanciare il carico di lavoro e ridurre al minimo lo spostamento dei dati da nodo a nodo. Per ulteriori informazioni, consultare Scelta del migliore stile di distribuzione.

Il caricamento di dati da file di testo sfrutta l'elaborazione parallela tramite la distribuzione del carico di lavoro tra più nodi, leggendo simultaneamente da più file. Per ulteriori informazioni su come caricare dati in tabelle, consultare Best practice di Amazon Redshift per il caricamento di dati.

Storage dei dati a colonne

Lo storage a colonne per le tabelle di database riduce drasticamente i requisiti complessivi di I/O su disco ed è un fattore importante nell'ottimizzazione delle prestazioni delle query di analisi. L'archiviazione a colonne delle informazioni delle tabelle di database riduce il numero di richieste di I/O su disco e la quantità di dati da caricare dal disco. Il caricamento di meno dati in memoria permette ad Amazon Redshift di eseguire più elaborazione in memoria durante l'esecuzione di query. Per una spiegazione più dettagliata, consultare Storage colonnare.

Quando le colonne sono ordinate nel modo appropriato, l'elaboratore di query è in grado di escludere rapidamente un subset di blocchi di dati di grandi dimensioni. Per ulteriori informazioni, consultare Scelta della migliore chiave di ordinamento.

Compressione dei dati

La compressione dei dati riduce i requisiti di storage, riducendo di conseguenza l'I/O su disco e migliorando le prestazioni delle query. Quando si esegue una query, i dati compressi vengono letti in memoria e decompressi durante l'esecuzione della query. Il caricamento di meno dati in memoria permette ad Amazon Redshift di allocare più memoria per analizzare i dati. Poiché lo storage a colonne archivia dati simili in sequenza, Amazon Redshift è in grado di applicare codifiche di compressione adattive appositamente associate a tipi di dati a colonne. Il modo migliore per abilitare la compressione dei dati su colonne di tabella è permettendo ad Amazon Redshift di

applicare codifiche di compressione ottimali durante il caricamento dei dati nella tabella. Per ulteriori informazioni sull'uso della compressione dei dati automatica, consultare Caricamento di tabelle con compressione automatica.

Ottimizzatore di query

Il motore di esecuzione di query di Amazon Redshift include un ottimizzatore di query compatibile con MPP e sfrutta inoltre l'archiviazione di dati basata su colonne. L'ottimizzatore di query di Amazon Redshift implementa estensioni e miglioramenti significativi per l'elaborazione di query di analisi complesse che includono spesso unioni di più tabelle, sottoquery e aggregazione. Per ulteriori informazioni sull'ottimizzazione delle query, consultare Ottimizzazione delle prestazioni delle query.

Caching dei risultati

Per ridurre il tempo di esecuzione delle query e migliorare le prestazioni del sistema, Amazon Redshift memorizza i risultati di certi tipi di query nella memoria cache del nodo principale. Quando un utente invia una query, Amazon Redshift controlla nella cache dei risultati l'eventuale presenza di una copia valida dei risultati della query nella cache. Se viene trovata una corrispondenza nella cache dei risultati, Amazon Redshift usa i risultati memorizzati nella cache e non esegue la query. Il caching dei risultati è trasparente all'utente.

Per impostazione predefinita, il caching dei risultati è abilitato. Per disattivare il caching dei risultati per la sessione corrente, imposta il parametro enable_result_cache_for_session su `off`.

Amazon Redshift usa i risultati nella cache per una nuova query quando si verificano tutte queste condizioni:

- L'utente che invia la query dispone dell'autorizzazione per accedere agli oggetti usati nella query.
- La tabella o le visualizzazioni nella query non sono state modificate.
- La query non usa una funzione che deve essere valutata a ogni esecuzione, come GETDATE.
- La query non fa riferimento a tabelle esterne Amazon Redshift Spectrum.
- I parametri di configurazione che potrebbero influire sui risultati della query sono invariati.
- La query corrisponde sintatticamente alla query nella cache.

Per massimizzare l'efficacia della cache e l'uso efficiente delle risorse, Amazon Redshift non memorizza nella cache alcuni set di risultati delle query di grandi dimensioni. Amazon Redshift determina se memorizzare nella cache i risultati delle query in base a diversi fattori. Questi fattori includono il numero di voci nella cache e il tipo di istanza del cluster di Amazon Redshift.

Per determinare se una query ha usato la cache dei risultati, eseguire una query sulla vista di sistema SVL_QLOG. Se una query ha usato la cache dei risultati, la colonna di query di origine restituisce l'ID query della query di origine. Se il caching dei risultati non è stato usato, il valore della colonna di query di origine è NULL.

L'esempio seguente mostra che le query inviate da userid 104 e userid 102 usano la cache dei risultati delle query eseguite da userid 100.

```
select userid, query, elapsed, source_query from svl_qlog
where userid > 1
order by query desc;

userid | query  | elapsed  | source_query
-------+--------+----------+-------------
   104 | 629035 |       27 |       628919
   104 | 629034 |       60 |       628900
   104 | 629033 |       23 |       628891
   102 | 629017 |  1229393 |
   102 | 628942 |       28 |       628919
   102 | 628941 |       57 |       628900
   102 | 628940 |       26 |       628891
   100 | 628919 | 84295686 |
   100 | 628900 | 87015637 |
   100 | 628891 | 58808694 |
```

Codice compilato

Il nodo principale distribuisce codice compilato completamente ottimizzato tra tutti i nodi di un cluster. La compilazione della query riduce la gestione associata a un interprete e di conseguenza aumenta la velocità di esecuzione, in particolare per le query complesse. Il codice compilato viene memorizzato nella cache e condiviso tra sessioni sullo stesso cluster. Di conseguenza, le esecuzioni future della stessa query saranno più veloci, spesso anche quando i parametri sono diversi.

Il motore di esecuzione delle query compila codice diverso per i protocolli di connessione JDBC e ODBC in modo che ognuno dei due client che usano protocolli diversi paghi i costi della prima compilazione del codice. Tuttavia, altri client che usano lo stesso protocollo godranno del vantaggio di poter condividere il codice nella cache.

Storage colonnare

L'archiviazione a colonne per le tabelle di database è un fattore importante nell'ottimizzazione delle prestazioni delle query di analisi, in quanto riduce drasticamente i requisiti complessivi di I/O su disco. Riduce la quantità di dati che è necessario caricare dal disco.

La serie di figure seguente descrive in che modo l'archiviazione di dati a colonne migliori l'efficienza e come questo si traduca in altrettanta efficienza durante il recupero di dati in memoria.

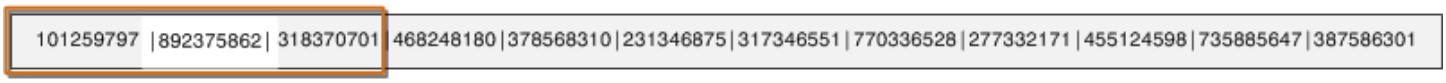
La prima figura mostra in che modo vengono generalmente archiviati i record delle tabelle di database in blocchi di dischi per riga.

SSN	Name	Age	Addr	City	St
101259797	SMITH	88	899 FIRST ST	JUNO	AL
892375862	CHIN	37	16137 MAIN ST	POMONA	CA
318370701	HANDU	12	42 JUNE ST	CHICAGO	IL

In una normale tabella di database relazionale, ogni riga contiene valori dei campi per un singolo record. Nello storage di database basato su righe, i blocchi di dati archiviano i valori in sequenza per ogni colonna consecutiva che costituisce l'intera riga. Se le dimensioni del blocco sono minori di quelle di un record, lo storage per un intero record può richiedere più di un blocco. Se le dimensioni del blocco sono maggiori di quelle di un record, lo storage per un intero record può richiedere meno di un blocco, causando un uso poco efficiente dello spazio su disco. Nelle applicazioni di elaborazione di transazioni online (OLTP), la maggior parte delle transazioni comporta letture e scritture frequenti di tutti i valori per interi record, in genere un record o un numero ridotto di record per volta. Di conseguenza, lo storage basato su righe è ottimale per i database OLTP.

La figura seguente mostra in che modo con lo storage a colonne i valori per ogni colonna vengono archiviati in sequenza in blocchi del disco.

SSN	Name	Age	Addr	City	St
101259797	SMITH	88	899 FIRST ST	JUNO	AL
892375862	CHIN	37	16137 MAIN ST	POMONA	CA
318370701	HANDU	12	42 JUNE ST	CHICAGO	IL



Block 1

Usando lo storage a colonne, ogni blocco di dati archivia i valori di una singola colonna per più righe. Man mano che i record raggiungono il sistema, Amazon Redshift converte in modo trasparente i dati in storage a colonne per ognuna delle colonne.

In questo esempio semplificato, usando lo storage a colonne, ogni blocco di dati contiene valori dei campi di colonna per tre volte il numero di record dello storage basato su righe. Questo significa che la lettura dello stesso numero di valori dei campi di colonna per lo stesso numero di record richiede un terzo delle operazioni di I/O rispetto allo storage basato su righe. In pratica, usando tabelle con quantità molto elevate di colonne e quantità molto elevate di righe, l'efficienza dello storage è ancora maggiore.

Un altro vantaggio è che, poiché ogni blocco contiene lo stesso tipo di dati, i dati dei blocchi possono usare uno schema di compressione selezionato appositamente per il tipo di dati di colonna, riducendo ulteriormente lo spazio su disco e l'I/O. Per ulteriori informazioni sulle codifiche di compressione in base ai tipi di dati, consultare Codifiche di compressione.

I risparmi in termini di spazio per lo storage dei dati su disco si applicano anche al recupero e al successivo storage dei dati in memoria. Poiché molte operazioni di database devono solo accedere o operare in una colonna o un numero ridotto di colonne per volta, puoi risparmiare spazio di memoria recuperando solo i blocchi per le colonne effettivamente necessarie per una query. Nei casi in cui le transazioni OLTP interessano in genere la maggior parte o tutte le colonne in una riga per un numero ridotto di record, le query del data warehouse leggono normalmente solo poche colonne per un numero molto elevato di righe. Questo significa che la lettura dello stesso numero di valori dei campi di colonna per lo stesso numero di righe richiede un numero inferiore di operazioni di I/O. Utilizza una quantità inferiore della memoria necessaria per l'elaborazione di blocchi per riga. In pratica, usando tabelle con quantità molto elevate di colonne e quantità molto elevate di righe, i vantaggi in termini di efficienza sono proporzionalmente maggiori. Ad esempio, supponi che una tabella contenga 100 colonne. Una query che usa cinque colonne dovrà leggere solo circa il 5% dei dati contenuti nella tabella. Questi vantaggi si ripetono possibilmente per miliardi o addirittura trilioni di record per i

database di grandi dimensioni. Al contrario, un database basato su righe dovrebbe leggere anche i blocchi che contengono le 95 colonne non necessarie.

In genere le dimensioni dei blocchi di database sono comprese tra 2 KB e 32 KB. In Amazon Redshift la dimensione dei blocchi è pari a 1 MB, che è più efficiente e riduce ulteriormente il numero di richieste di I/O necessarie per eseguire qualsiasi caricamento di database o altre operazioni che fanno parte dell'esecuzione di query.

Gestione dei carichi di lavoro

La gestione dei carichi di lavoro in Amazon Redshift permette agli utenti di affrontare in modo flessibile le priorità all'interno dei carichi di lavoro in modo da evitare che query brevi a esecuzione rapida non restino bloccate in code dietro query a esecuzione prolungata.

La gestione dei carichi di lavoro in Amazon Redshift WLM crea code di query in fase di runtime in base alle classi di servizio, che definiscono i parametri di configurazione per diversi tipi di query, tra cui code di sistema interne e code accessibili dall'utente. Dal punto di vista dell'utente, una classe di servizio accessibile dall'utente e una coda sono funzionalmente equivalenti. Per coerenza, questa documentazione usa il termine coda con il significato di classe di servizio accessibile dall'utente e di coda di runtime.

Quando si esegue una query, la gestione dei carichi di lavoro assegna la query a una coda in base al gruppo di utenti dell'utente o associando un gruppo di query elencato nella configurazione della coda a un'etichetta del gruppo di query impostata dall'utente in fase di runtime.

> Attualmente, l'impostazione predefinita per i cluster che utilizzano il gruppo di parametri predefini to è di utilizzare il WLM automatico. Il WLM automatico gestisce le query simultanee e l'allocazione della memoria. Per ulteriori informazioni, consultare Implementazione del WLM automatico.

Con WLM manuale, Amazon Redshift configura una coda con un livello di simultaneità cinque, che permette l'esecuzione simultanea di un massimo di cinque query, più una coda dell'utente con privilegi avanzati predefinita, con un livello di simultaneità uno. Puoi definire fino a otto code. Ogni coda può essere configurata con un livello massimo di simultaneità pari a 50. Il livello massimo di simultaneità totale per tutte le code definite dall'utente (esclusa la coda dell'utente con privilegi avanzati) è 50.

Il modo più semplice per modificare la configurazione della gestione dei carichi di lavoro consiste nell'usare la Console di gestione di Amazon Redshift. Inoltre è possibile utilizzare l'interfaccia a riga di comando (CLI) di Amazon Redshift o l'API di Amazon Redshift.

Per ulteriori informazioni sull'implementazione e sull'uso della gestione dei carichi di lavoro, consultare Implementazione della gestione del carico di lavoro.

Utilizzo di Amazon Redshift con altri servizi

Amazon Redshift si integra con altri AWS servizi per consentirti di spostare, trasformare e caricare i dati in modo rapido e affidabile, utilizzando funzionalità di sicurezza dei dati.

Spostamento di dati tra Amazon Redshift e Amazon S3

Amazon Simple Storage Service (Amazon S3) è un servizio Web che consente di memorizzare dati nel cloud. Amazon Redshift sfrutta l'elaborazione parallela per leggere e caricare dati da più file di dati archiviati nei bucket Amazon S3. Per ulteriori informazioni, consultare Caricamento di dati da Amazon S3.

È possibile usare l'elaborazione parallela anche per esportare dati dal data warehouse di Amazon Redshift in più file di dati in Amazon S3. Per ulteriori informazioni, consultare Scaricamento dei dati.

Utilizzo di Amazon Redshift con Amazon DynamoDB

Amazon DynamoDB è un servizio di database NoSQL completamente gestito. È possibile usare il comando COPY per caricare in una tabella di Amazon Redshift dati provenienti da una singola tabella Amazon DynamoDB. Per ulteriori informazioni, consultare Caricamento di dati da una tabella Amazon DynamoDB.

Importazione di dati da host remoti tramite SSH

È possibile usare il comando COPY in Amazon Redshift per caricare dati da uno o più host remoti, come cluster Amazon EMR, istanze Amazon EC2 o altri computer. COPY si connette agli host remoti utilizzando SSH ed esegue comandi sugli host remoti per generare dati. Amazon Redshift supporta più connessioni simultanee. Il comando COPY legge e carica l'output da più origini host in parallelo. Per ulteriori informazioni, consulta Caricamento di dati da host remoti.

Automatizzazione dei caricamenti di dati utilizzando AWS Data Pipeline

Puoi utilizzarlo AWS Data Pipeline per automatizzare lo spostamento e la trasformazione dei dati da e verso Amazon Redshift. Utilizzando le funzionalità di pianificazione integrate di AWS Data

Pipeline, puoi pianificare ed eseguire lavori ricorrenti senza dover scrivere una complessa logica di trasferimento o trasformazione dei dati. Ad esempio, è possibile configurare un processo ricorrente per la copia automatica di dati da Amazon DynamoDB ad Amazon Redshift. Per un tutorial che illustra il processo di creazione di una pipeline che sposta periodicamente i dati da Amazon S3 ad Amazon Redshift, consulta Copiare i dati su Amazon Redshift utilizzando nella Developer Guide. AWS Data Pipeline AWS Data Pipeline

Migrazione dei dati utilizzando () AWS Database Migration ServiceAWS DMS

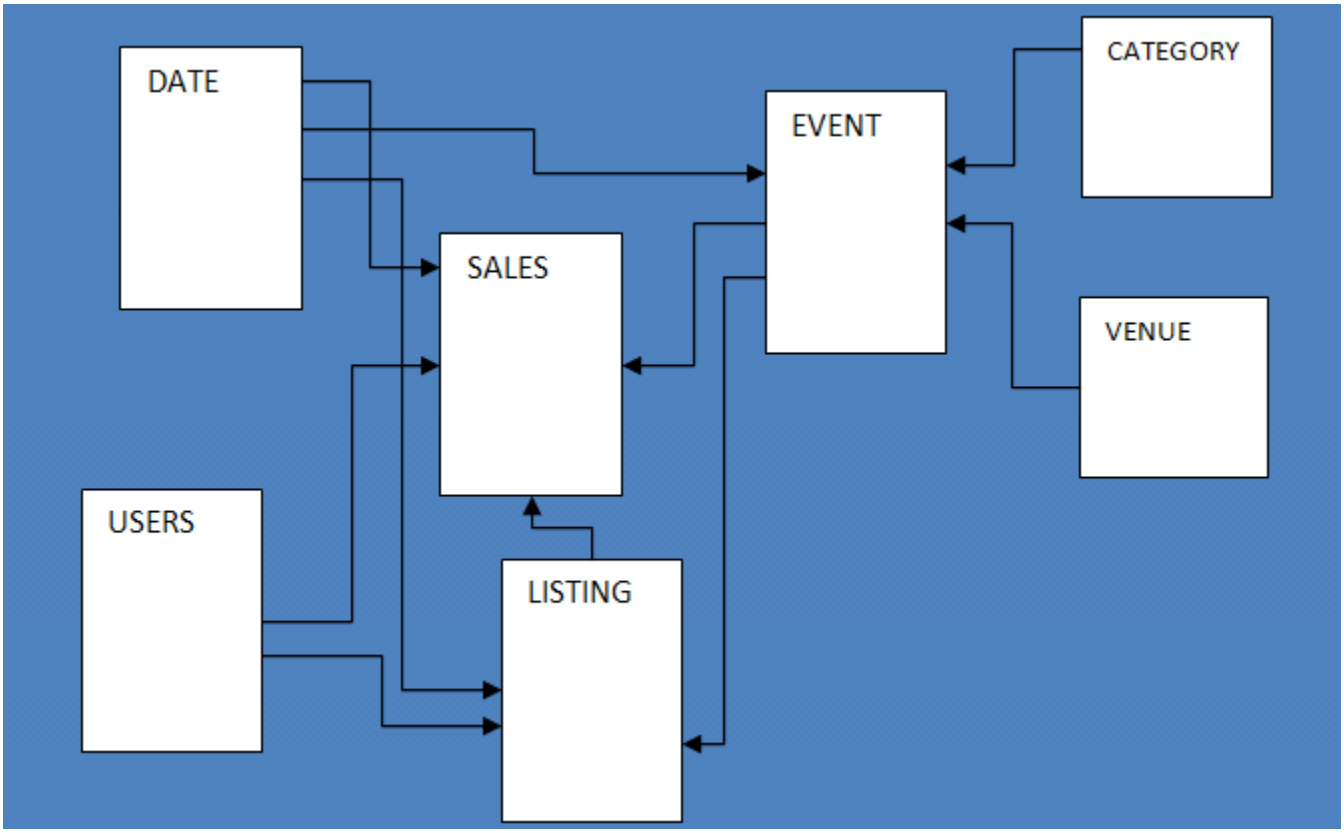
Puoi migrare i dati su Amazon AWS Database Migration Service Redshift utilizzando. AWS DMS può migrare i dati da e verso i database commerciali e open source più utilizzati come Oracle, PostgreSQL, Microsoft SQL Server, Amazon Redshift, Aurora DB cluster, DynamoDB, Amazon S3, MariaDB e MySQL. Per ulteriori informazioni, consultare Utilizzo di un database Amazon Redshift come destinazione per AWS Database Migration Service.

Database di esempio

Argomenti

- Tabella CATEGORY
- Tabella DATE
- Tabella EVENT
- Tabella VENUE
- Tabella USERS
- Tabella LISTING
- Tabella SALES

Nella maggior parte degli esempi della documentazione di Amazon Redshift viene utilizzato un database di esempio chiamato TICKIT. Questo piccolo database è composto da sette tabelle: due tabelle di fatti e cinque di dimensioni. Puoi caricare il set di dati TICKIT seguendo i passaggi indicati nella Fase 4: Caricare i dati da Amazon S3 ad Amazon Redshift nella Amazon Redshift Getting Started Guide.



Questa applicazione del database di esempio consente agli analisti di tenere traccia delle attività di vendita per il sito Web fittizio TICKIT in cui gli utenti acquistano e vendono biglietti online per eventi sportivi, spettacoli e concerti. In particolare, gli analisti possono monitorare il movimento dei biglietti nel tempo, le percentuali di successo dei venditori e le stagioni, i luoghi e gli eventi più venduti. Gli analisti possono utilizzare queste informazioni per offrire incentivi agli acquirenti e ai venditori che visitano il sito, attirare nuovi utenti e sostenere pubblicità e promozioni.

Ad esempio, la seguente query restituisce i migliori cinque venditori di San Diego, in base al numero di biglietti venduti nel 2008:

```
select sellerid, username, (firstname ||' '|| lastname) as name,
city, sum(qtysold)
from sales, date, users
where sales.sellerid = users.userid
and sales.dateid = date.dateid
and year = 2008
and city = 'San Diego'
group by sellerid, username, name, city
order by 5 desc
limit 5;
```

```
sellerid | username |       name        |   city    | sum
---------+----------+-------------------+-----------+-----
49977 | JJK84WTE | Julie Hanson      | San Diego |  22
19750 | AAS23BDR | Charity Zimmerman | San Diego |  21
29069 | SVL81MEQ | Axel Grant        | San Diego |  17
43632 | VAG08HKW | Griffin Dodson    | San Diego |  16
36712 | RXT40MKU | Hiram Turner      | San Diego |  14
(5 rows)
```

Il database utilizzato per gli esempi in questa guida contiene un piccolo set di dati, le due tabelle dei fatti contengono ciascuna meno di 200.000 righe e le dimensioni variano da 11 righe nella tabella CATEGORY fino a circa 50.000 righe nella tabella USERS.

In particolare, gli esempi del database in questa guida dimostrano le funzionalità chiave della progettazione della tabella Amazon Redshift:

- Distribuzione dei dati
- Ordinamento dei dati
- Compressione delle colonne

Tabella CATEGORY

Nome colonna	Tipo di dati	Descrizione
CATID	SMALLINT	Chiave primaria, un valore ID univoco per ogni riga. Ogni riga rappresenta un tipo specifico di evento per cui i biglietti sono acquistati e venduti.
CATGROUP	VARCHAR(10)	Nome descrittivo per un gruppo di eventi, ad esempio **`Shows`** e **`Sports`**.
CATNAME	VARCHAR(10)	Breve nome descrittivo per un tipo di evento all'interno di un gruppo, ad esempio **`Opera`** e **`Musicals`**.
CATDESC	VARCHAR(50)	Nome descrittivo più lungo per il tipo di evento, ad esempio **`Musical theatre`**.

Tabella DATE

Nome colonna	Tipo di dati	Descrizione
DATEID	SMALLINT	Chiave primaria, un valore ID univoco per ogni riga. Ogni riga rappresenta un giorno dell'anno di calendario.
CALDATE	DATE	Data di calendario, ad esempio **2008-06-24** .
GIORNO	CHAR(3)	Giorno della settimana (forma breve), ad esempio **SA**.
WEEK	SMALLINT	Numero della settimana, ad esempio **26**.
MESE	CHAR(5)	Nome del mese (forma breve), ad esempio **JUN**.
QTR	CHAR(5)	Numero del trimestre (da **1** a **4**).
ANNO	SMALLINT	Anno a quattro cifre (**2008**).
HOLIDAY	BOOLEAN	Il flag che indica se il giorno è festivo (Stati Uniti).

Tabella EVENT

Nome colonna	Tipo di dati	Descrizione
EVENTID	INTEGER	Chiave primaria, un valore ID univoco per ogni riga. Ogni riga rappresenta un evento separato che si svolge in un luogo specifico in un determinato momento.
VENUEID	SMALLINT	Riferimento di chiave esterna alla tabella VENUE.
CATID	SMALLINT	Riferimento di chiave esterna alla tabella CATEGORY.
DATEID	SMALLINT	Riferimento di chiave esterna alla tabella DATE.
EVENTNAME	VARCHAR(200)	Nome dell'evento, ad esempio **Hamlet** o **La Traviata**.

Nome colonna	Tipo di dati	Descrizione
STARTTIME	TIMESTAMP	Data completa e ora di inizio dell'evento, ad esempio **`2008-10-10 19:30:00`** .

Tabella VENUE

Nome colonna	Tipo di dati	Descrizione
VENUEID	SMALLINT	Chiave primaria, un valore ID univoco per ogni riga. Ogni riga rappresenta un luogo specifico in cui si svolgono gli eventi.
VENUENAME	VARCHAR(100)	Nome esatto della sede, ad esempio **`Cleveland Browns Stadium`**.
VENUECITY	VARCHAR(30)	Nome della città, ad esempio **`Cleveland`** .
VENUESTATE	CHAR(2)	Abbreviazione dello stato o della provincia composta da due lettere (Stati Uniti e Canada), ad esempio **`OH`**.
VENUESEATS	INTEGER	Numero massimo di posti a sedere disponibili presso il luogo, se noti, ad esempio **`73200`**. A scopo dimostrativo, questa colonna contiene valori nulli e zero.

Tabella USERS

Nome colonna	Tipo di dati	Descrizione
USERID	INTEGER	Chiave primaria, un valore ID univoco per ogni riga. Ogni riga rappresenta un utente registrato (un acquirente o un venditore o entrambi) che ha proposto o acquistato i biglietti per almeno un evento.

Nome colonna	Tipo di dati	Descrizione
USERNAME	CHAR(8)	Un nome utente alfanumerico composto da 8 caratteri , ad esempio **PGL08LJI**.
FIRSTNAME	VARCHAR(30)	Il nome dell'utente, ad esempio **Victor**.
LASTNAME	VARCHAR(30)	Il cognome dell'utente, ad esempio **Hernandez** .
CITY	VARCHAR(30)	La città di origine dell'utente, ad esempio **Napervill e** .
STATE	CHAR(2)	Il paese di origine dell'utente, ad esempio **GA**.
EMAIL	VARCHAR(100)	L'indirizzo e-mail dell'utente; questa colonna contiene valori latini casuali, ad esempio **turpis@ac cumsanlaoreet.org** .
PHONE	CHAR(14)	Il numero di telefono dell'utente composto da 14 caratteri, ad esempio **(818) 765-4255**.
LIKESPORT S, ...	BOOLEAN	Una serie di 10 diverse colonne che identificano i "mi piace" e i "non mi piace" dell'utente con i valori **true** e **false**.

Tabella LISTING

Nome colonna	Tipo di dati	Descrizione
LISTID	INTEGER	Chiave primaria, un valore ID univoco per ogni riga. Ogni riga rappresenta l'elenco di un lotto di biglietti per un evento specifico.
SELLERID	INTEGER	Riferimento di chiave esterna alla tabella USERS che identifica l'utente che sta vendendo i biglietti.
EVENTID	INTEGER	Riferimento di chiave esterna alla tabella EVENT.

Nome colonna	Tipo di dati	Descrizione
DATEID	SMALLINT	Riferimento di chiave esterna alla tabella DATE.
NUMTICKETS	SMALLINT	Il numero di biglietti disponibili per la vendita, ad esempio **2** o **20**.
PRICEPERTICKET	DECIMAL(8,2)	Il prezzo fisso di un singolo biglietto, ad esempio **27.00** o **206.00**.
TOTALPRICE	DECIMAL(8,2)	Il prezzo totale per questo elenco (NUMTICKETS*PRICEPERTICKET).
LISTTIME	TIMESTAMP	La data completa e l'ora di pubblicazione dell'elenco, ad esempio **2008-03-18 07:19:35** .

Tabella SALES

Nome colonna	Tipo di dati	Descrizione
SALESID	INTEGER	Chiave primaria, un valore ID univoco per ogni riga. Ogni riga rappresenta la vendita di uno o più biglietti per un evento specifico, come viene offerto in un elenco specifico.
LISTID	INTEGER	Riferimento di chiave esterna alla tabella LISTING.
SELLERID	INTEGER	Riferimento di chiave esterna alla tabella USERS (l'utente che ha venduto i biglietti).
BUYERID	INTEGER	Riferimento di chiave esterna alla tabella USERS (l'utente che ha acquistato i biglietti).
EVENTID	INTEGER	Riferimento di chiave esterna alla tabella EVENT.
DATEID	SMALLINT	Riferimento di chiave esterna alla tabella DATE.

Nome colonna	Tipo di dati	Descrizione
QTYSOLD	SMALLINT	Il numero di biglietti venduti, da **1** a **8**. È possibile vendere un massimo di 8 biglietti in un'unica transazione.
PRICEPAID	DECIMAL(8,2)	Il prezzo totale pagato per i biglietti, ad esempio **75.00** o **488.00**. Il prezzo di un singolo biglietto corrisponde a PRICEPAID/QTYSOLD.
COMMISSION	DECIMAL(8,2)	La commissione del 15% che l'azienda riscuote sulla vendita, ad esempio **11.25** o **73.20**. Il venditore riceve l'85% del valore di PRICEPAID.
SALETIME	TIMESTAMP	La data completa e l'ora di completamento della vendita, ad esempio **2008-05-24 06:21:47** .

Best practice di Amazon Redshift

Nelle pagine seguenti vengono descritte le best practice per pianificare un proof of concept, progettare tabelle, caricare dati in tabelle e scrivere query per Amazon Redshift e vengono fornite informazioni sull'utilizzo di Amazon Redshift Advisor.

Amazon Redshift è differente dagli altri sistemi di database SQL. Per comprendere appieno i vantaggi dell'architettura di Amazon Redshift, è necessario progettare, creare e caricare tabelle allo scopo di utilizzare l'elaborazione MPP (Massively Parallel Processing), l'archiviazione dei dati a colonne e la compressione di dati a colonne. Se i tempi di caricamento di dati e di esecuzione di query sono più lunghi del previsto o di quanto desideri, è possibile che non hai preso in considerazione informazioni importanti.

Uno sviluppatore di database SQL esperto può leggere questo argomento prima di iniziare lo sviluppo di data warehouse Amazon Redshift.

Se invece sei alle prime armi in fatto di sviluppo di database SQL, questo argomento non è il modo migliore di cominciare. Ti consigliamo di iniziare leggendo Common database tasks e provando tu stesso gli esempi.

In questo argomento troverai una panoramica dei principi di sviluppo più importanti nonché specifici suggerimenti, esempi e best practice per l'implementazione di tali principi. Non esiste una best practice valida per tutte le applicazioni, Valuta tutte le opzioni a tua disposizione prima di finalizzare la progettazione di un database. Per ulteriori informazioni, consultare Utilizzo dell'ottimizzazione automatica delle tabelle, Caricamento dei dati, Ottimizzazione delle prestazioni delle query e i capitoli di riferimento.

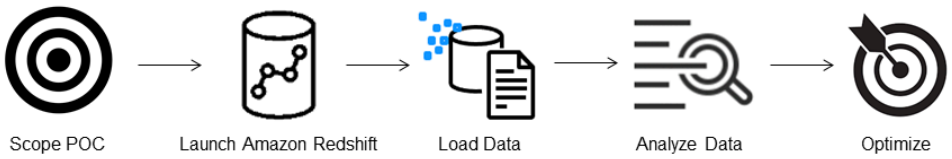
Argomenti

Esegui un proof of concept (POC) per Amazon Redshift

Amazon Redshift è un popolare data warehouse su cloud, che offre un servizio basato sul cloud completamente gestito che si integra con il data lake Amazon Simple Storage Service di un'organizzazione, flussi di lavoro in tempo reale, flussi di lavoro di machine learning (ML), flussi di lavoro transazionali e molto altro. Le seguenti sezioni ti guidano nel processo di creazione di un proof of concept (POC) su Amazon Redshift. Le informazioni qui riportate ti aiutano a fissare obiettivi per il tuo POC e sfruttano gli strumenti in grado di automatizzare il provisioning e la configurazione dei servizi per il tuo POC.

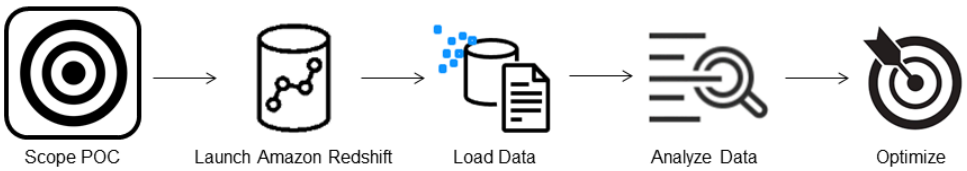
> **Note**
>
> Per una copia di queste informazioni in formato PDF, scegli il link Esegui il tuo POC Redshift nella pagina delle risorse di Amazon Redshift.

Quando esegui un POC di Amazon Redshift, esegui test, dimostri e adotti funzionalità che vanno best-in-class dalle funzionalità di sicurezza, alla scalabilità elastica, alla facilità di integrazione e inserimento e alle opzioni flessibili di architettura dei dati decentralizzata.



Segui questi passaggi per condurre un POC di successo.

Fase 1: Definisci l'ambito del POC



Quando si esegue un POC, è possibile scegliere di utilizzare i propri dati oppure utilizzare set di dati di benchmarking. Quando scegli i tuoi dati, esegui le tue query sui dati. Con i dati di benchmarking,

insieme al benchmark vengono fornite interrogazioni di esempio. Vedi Utilizzare set di dati di esempio per maggiori dettagli se non sei ancora pronto a condurre un POC con i tuoi dati.

In generale, consigliamo di utilizzare due settimane di dati per un POC Amazon Redshift.

Inizia effettuando le seguenti operazioni:

1. Identifica i requisiti aziendali e funzionali, quindi procedi a ritroso. Esempi comuni sono: prestazioni più veloci, costi inferiori, test di un nuovo carico di lavoro o funzionalità o confronto tra Amazon Redshift e un altro data warehouse.
2. Stabilisci obiettivi specifici che diventino i criteri di successo per il POC. Ad esempio, partendo da prestazioni più elevate, crea un elenco dei primi cinque processi che desideri accelerare e includi i tempi di esecuzione correnti insieme al tempo di esecuzione richiesto. Questi possono essere report, interrogazioni, processi ETL, acquisizione di dati o qualsiasi altro sia il vostro attuale problema.
3. Identifica l'ambito e gli artefatti specifici necessari per eseguire i test. Di quali set di dati hai bisogno per migrare o importare continuamente in Amazon Redshift e quali query e processi sono necessari per eseguire i test di misurazione in base ai criteri di successo? Ci sono due modi per effettuare questa operazione:

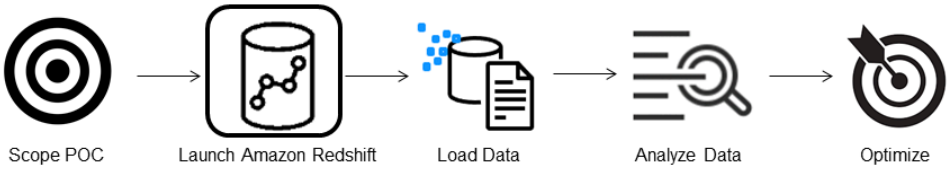
   Porta i tuoi dati

   - Per testare i tuoi dati, crea l'elenco minimo valido di artefatti di dati necessario per verificare i tuoi criteri di successo. Ad esempio, se il data warehouse corrente ha 200 tabelle, ma i report che desideri testare ne richiedono solo 20, il POC può essere eseguito più velocemente utilizzando solo il sottoinsieme di tabelle più piccolo.

   Utilizza set di dati di esempio

   - Se non disponi di set di dati personalizzati, puoi comunque iniziare a creare un POC su Amazon Redshift utilizzando i set di dati di benchmark standard del settore come TPC-DS o TPC-H ed eseguire query di benchmarking di esempio per sfruttare la potenza di Amazon Redshift. È possibile accedere a questi set di dati dall'interno del data warehouse Amazon Redshift dopo la creazione. Per istruzioni dettagliate su come accedere a questi set di dati e alle query di esempio, consulta. Fase 2: Avvia Amazon Redshift

Fase 2: Avvia Amazon Redshift



Amazon Redshift accelera i tempi di acquisizione delle informazioni con un data warehousing cloud rapido, facile e sicuro su larga scala. Puoi iniziare rapidamente avviando il tuo warehouse sulla console Redshift Serverless e passare dai dati agli approfondimenti in pochi secondi. Con Redshift Serverless, puoi concentrarti sul raggiungimento dei tuoi risultati di business senza preoccuparti della gestione del tuo data warehouse.

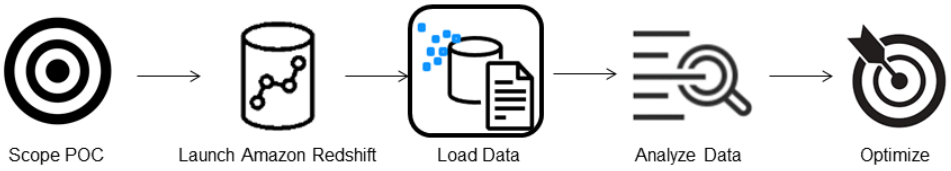
Configura Amazon Redshift Serverless

La prima volta che utilizzi Redshift Serverless, la console ti guida attraverso i passaggi necessari per avviare il tuo warehouse. Potresti anche avere diritto a un credito per l'utilizzo di Redshift Serverless nel tuo account. Per ulteriori informazioni sulla scelta di una prova gratuita, consultare Prova gratuita di Amazon Redshift. Segui i passaggi indicati nella Creazione di un data warehouse con Redshift Serverless nella Guida introduttiva di Amazon Redshift per creare un data warehouse con Redshift Serverless. Se non disponi di un set di dati da caricare, la guida contiene anche passaggi su come caricare un set di dati di esempio.

Se hai già avviato Redshift Serverless nel tuo account, segui i passaggi in Creazione di un gruppo di lavoro con uno spazio dei nomi nella Amazon Redshift Management Guide. Una volta che il tuo magazzino sarà disponibile, puoi scegliere di caricare i dati di esempio disponibili in Amazon Redshift. Per informazioni sull'utilizzo di Amazon Redshift Query Editor v2 per caricare i dati, consulta Loading sample data nella Amazon Redshift Management Guide.

Se intendi importare i tuoi dati anziché caricare il set di dati di esempio, consulta. Fase 3: Caricare i dati

Fase 3: Caricare i dati



Dopo aver avviato Redshift Serverless, il passaggio successivo consiste nel caricare i dati per il POC. Che tu stia caricando un semplice file CSV, importando dati semistrutturati da S3 o trasmettendo dati direttamente in streaming, Amazon Redshift offre la flessibilità necessaria per spostare rapidamente e facilmente i dati nelle tabelle Amazon Redshift dalla fonte.

Scegli uno dei seguenti metodi per caricare i dati.

Carica un file locale

Per un'acquisizione e un'analisi rapide, puoi utilizzare Amazon Redshift Query Editor v2 per caricare facilmente i file di dati dal desktop locale. Ha la capacità di elaborare file in vari formati come CSV, JSON, AVRO, PARQUET, ORC e altri. Per consentire ai tuoi utenti, in qualità di amministratore, di caricare dati da un desktop locale utilizzando l'editor di query v2, devi specificare un bucket Amazon S3 comune e l'account utente deve essere configurato con le autorizzazioni appropriate. Puoi seguire il caricamento dei dati reso semplice e sicuro in Amazon Redshift utilizzando Query Editor V2 come guida. step-by-step

Caricare un file Amazon S3

Per caricare dati da un bucket Amazon S3 in Amazon Redshift, inizia a utilizzare il comando COPY, specificando la posizione Amazon S3 di origine e la tabella Amazon Redshift di destinazione. Assicurati che i ruoli e le autorizzazioni IAM siano configurati correttamente per consentire ad Amazon Redshift l'accesso al bucket Amazon S3 designato. Segui il tutorial: Caricamento dei dati da Amazon S3 come guida. step-by-step Puoi anche scegliere l'opzione Carica dati nell'editor di query v2 per caricare direttamente i dati dal tuo bucket S3.

Inserimento continuo dei dati

Autocopy (in anteprima) è un'estensione del comando COPY e automatizza il caricamento continuo dei dati dai bucket Amazon S3. Quando crei un processo di copia, Amazon Redshift rileva quando vengono creati nuovi file Amazon S3 in un percorso specificato e quindi li carica automaticamente

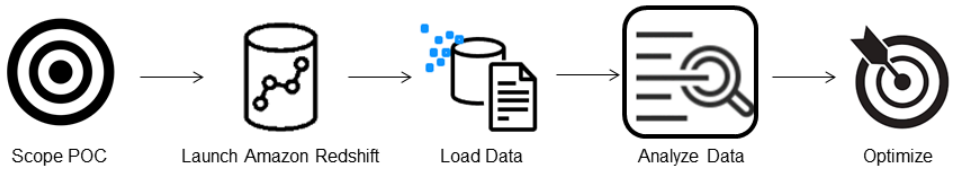
senza il tuo intervento. Amazon Redshift tiene traccia dei file caricati per verificare che vengano caricati una sola volta. Per istruzioni su come creare lavori di copia, consulta COPY JOB (anteprima)

> **Note**
>
> La copia automatica è attualmente in anteprima e supportata solo in cluster predisposti in determinati casi. Regioni AWS Per creare un cluster di anteprima per la copia automatica, consulta. Importazione continua di file da Amazon S3 (anteprima)

Carica i tuoi dati di streaming

L'ingestione di streaming consente l'inserimento a bassa latenza e alta velocità di dati di flusso da Amazon Kinesis Data Streams e Amazon Managed Streaming for Apache Kafka in Amazon Redshift. L'inserimento dello streaming di Amazon Redshift utilizza una vista materializzata, che viene aggiornata direttamente dallo stream utilizzando l'aggiornamento automatico. La vista materializzata viene mappata all'origine dati del flusso. Puoi eseguire filtri e aggregazioni sui dati del flusso come parte della definizione della vista materializzata. Per step-by-step indicazioni su come caricare dati da uno stream, consulta la sezione Guida introduttiva ad Amazon Kinesis Data Streams o Guida introduttiva ad Amazon Managed Streaming for Apache Kafka.

Fase 4: Analizza i tuoi dati



Dopo aver creato il gruppo di lavoro e lo spazio dei nomi Redshift Serverless e aver caricato i dati, puoi eseguire immediatamente le query aprendo l'editor di query v2 dal pannello di navigazione della console Redshift Serverless. È possibile utilizzare l'editor di query v2 per testare la funzionalità delle query o le prestazioni delle query rispetto ai propri set di dati.

Esegui query con Amazon Redshift Query Editor v2

Puoi accedere all'editor di query v2 dalla console Amazon Redshift. Consulta Semplifica l'analisi dei dati con Amazon Redshift Query Editor v2 per una guida completa su come configurare, connettere ed eseguire query con Query Editor v2.

In alternativa, se desideri eseguire un test di carico come parte del tuo POC, puoi farlo seguendo i seguenti passaggi per installare ed eseguire Apache JMeter.

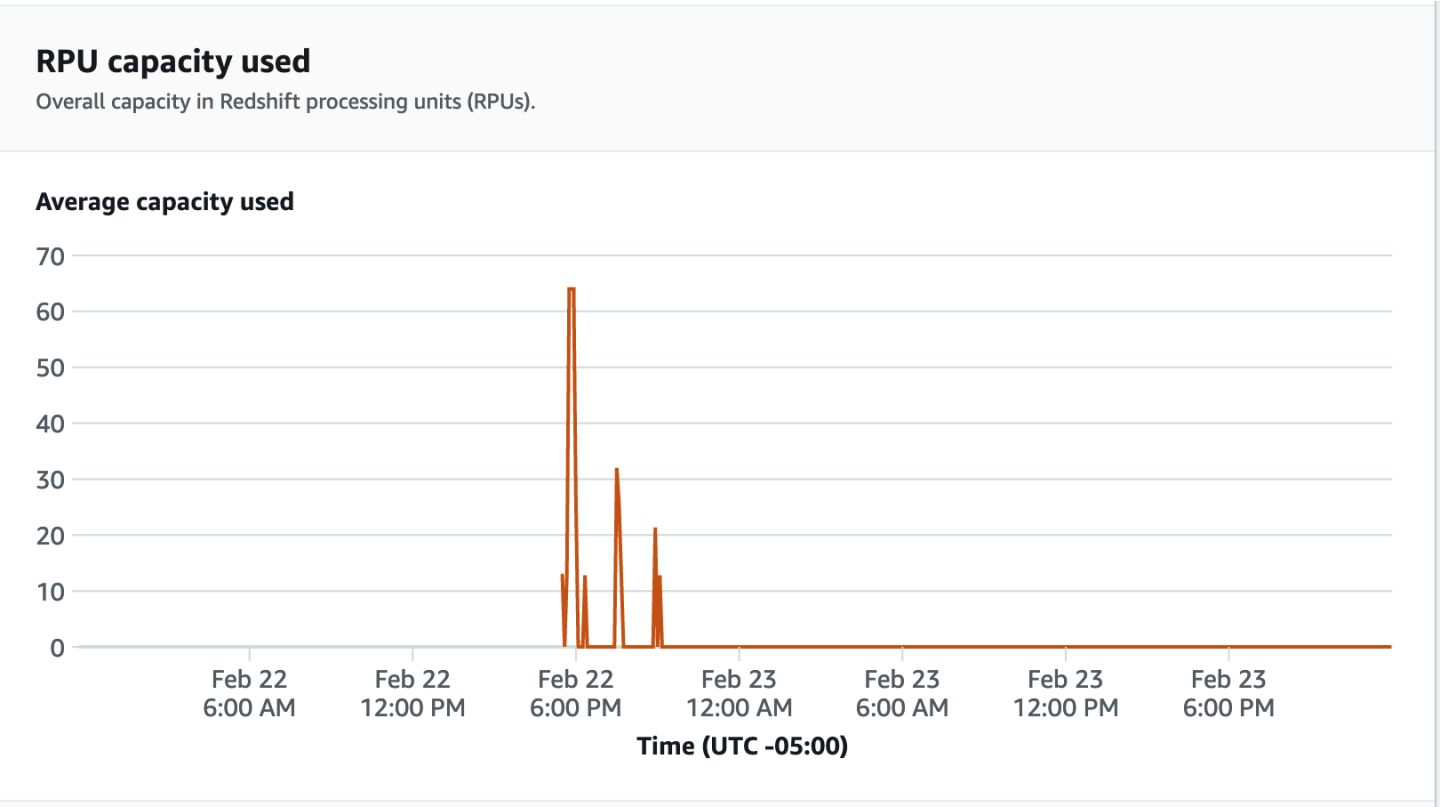
Esegui un test di carico utilizzando Apache JMeter

Per eseguire un test di carico per simulare «N» utenti che inviano query contemporaneamente ad Amazon Redshift, puoi utilizzare Apache JMeter, uno strumento open source basato su Java.

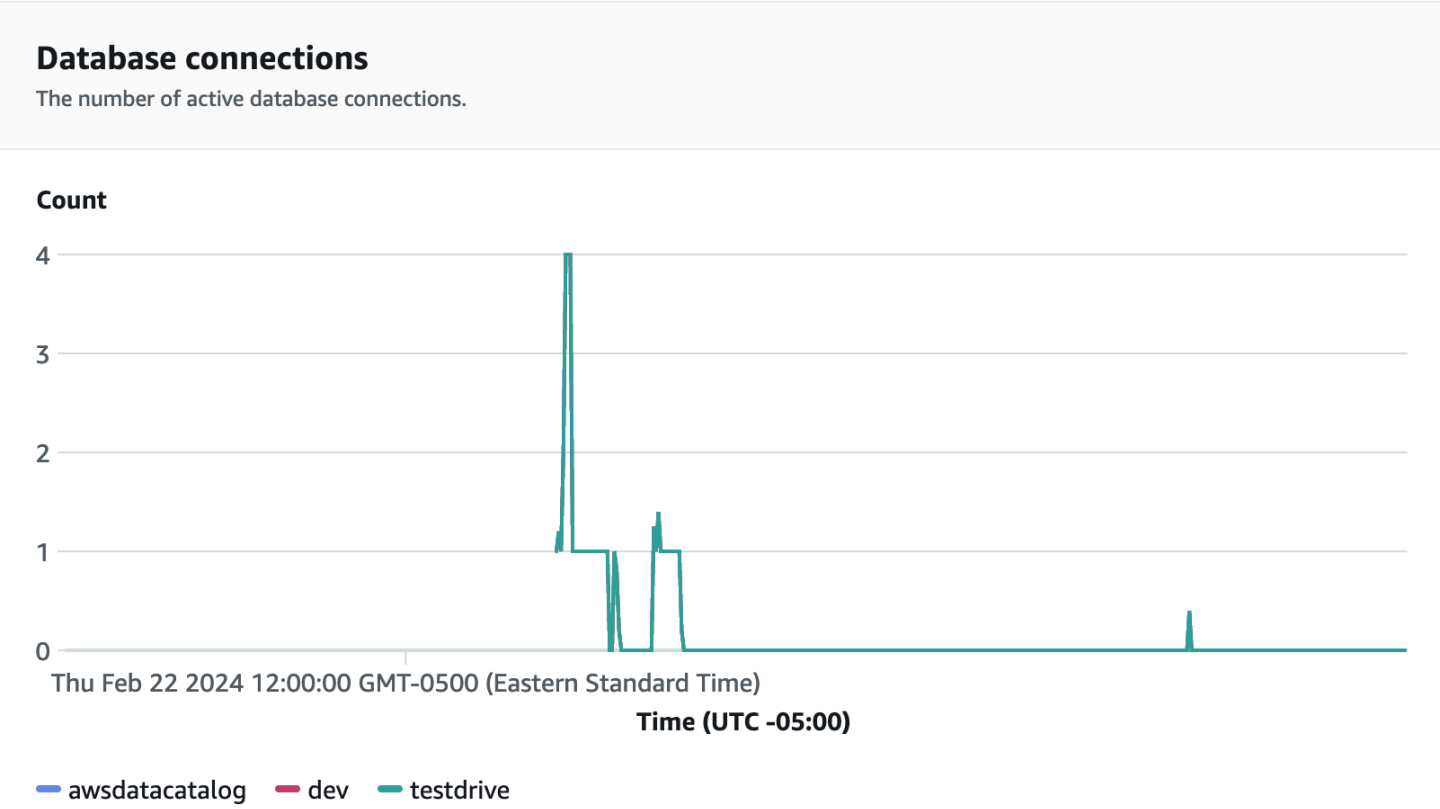
Per installare e configurare Apache JMeter per l'esecuzione sul tuo gruppo di lavoro Redshift Serverless, segui le istruzioni in Automatizza i test di carico di Amazon Redshift con Analytics Automation Toolkit. AWS Utilizza il toolkit AWS Analytics Automation (AAA), un'utilità open source per l'implementazione dinamica delle soluzioni Redshift, per avviare automaticamente queste risorse. Se hai caricato i tuoi dati in Amazon Redshift, assicurati di eseguire l'opzione Step #5 — Customize SQL, per assicurarti di fornire le istruzioni SQL appropriate che desideri testare sulle tue tabelle. Prova ognuna di queste istruzioni SQL una sola volta utilizzando l'editor di query v2 per assicurarti che vengano eseguite senza errori.

Dopo aver completato la personalizzazione delle istruzioni SQL e la finalizzazione del piano di test, salva ed esegui il piano di test sul gruppo di lavoro Redshift Serverless. Per monitorare l'avanzamento del test, apri la console Redshift Serverless, vai a Monitoraggio delle query e del database, scegli la scheda Cronologia delle query e visualizza le informazioni sulle tue query.

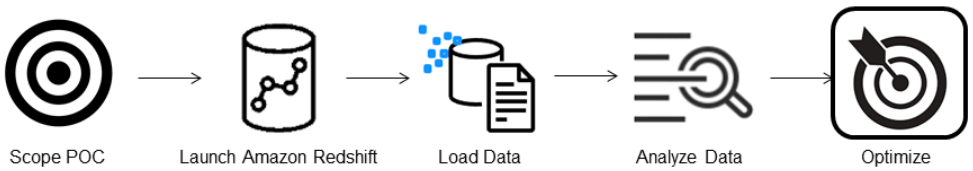
Per le metriche delle prestazioni, scegli la scheda Prestazioni del database sulla console Redshift Serverless, per monitorare metriche come le connessioni al database e l'utilizzo della CPU. Qui puoi visualizzare un grafico per monitorare la capacità RPU utilizzata e osservare come Redshift Serverless si ridimensiona automaticamente per soddisfare le richieste simultanee di carichi di lavoro mentre il test di carico è in esecuzione sul tuo gruppo di lavoro.



Le connessioni al database sono un'altra metrica utile da monitorare durante l'esecuzione del test di carico per vedere come il gruppo di lavoro gestisce numerose connessioni simultanee in un dato momento per soddisfare le crescenti richieste di carico di lavoro.



Fase 5: Ottimizzazione



Amazon Redshift consente a decine di migliaia di utenti di elaborare exabyte di dati ogni giorno e potenziare i propri carichi di lavoro di analisi offrendo una varietà di configurazioni e funzionalità per supportare casi d'uso individuali. Quando scelgono tra queste opzioni, i clienti cercano strumenti che li aiutino a determinare la configurazione di data warehouse più ottimale per supportare il carico di lavoro di Amazon Redshift.

Prova su strada

Puoi utilizzare Test Drive per riprodurre automaticamente il carico di lavoro esistente su potenziali configurazioni e analizzare gli output corrispondenti per valutare l'obiettivo ottimale verso cui migrare il carico di lavoro. Per informazioni sull'utilizzo di Test Drive per valutare diverse configurazioni

Amazon Redshift, consulta Trova la configurazione Amazon Redshift migliore per il tuo carico di lavoro utilizzando Redshift Test Drive.

Best practice di Amazon Redshift per la progettazione di tabelle

Durante la pianificazione di un database, alcune decisioni chiave in materia di progettazione di tabelle influenzano considerevolmente le prestazioni globali delle query. Queste scelte hanno inoltre un notevole effetto sui requisiti di storage, i quali alterano le prestazioni delle query diminuendo il numero di operazioni di I/O e riducendo al minimo la memoria necessaria per elaborare le query.

In questa sezione, è possibile trovare un riepilogo delle decisioni di progettazione più importanti e le best practice per l'ottimizzazione delle prestazioni delle query. Utilizzo dell'ottimizzazione automatica delle tabelle fornisce descrizioni ed esempi più dettagliati delle opzioni di progettazione delle tabelle.

Argomenti

Scelta della migliore chiave di ordinamento

Amazon Redshift archivia i dati sul disco nell'ordine stabilito dalla chiave di ordinamento. L'ottimizzatore di query Amazon Redshift utilizza tale ordinamento per determinare i piani di query ottimali.

> **Note**
>
> Quando si utilizza l'ottimizzazione automatica della tabella, non è necessario scegliere la chiave di ordinamento della tabella. Per ulteriori informazioni, consultare Utilizzo dell'ottimizzazione automatica delle tabelle.

Seguono alcuni suggerimenti per l'approccio migliore:

- Per fare in modo che Amazon Redshift scelga l'ordinamento appropriato, specificare AUTO per la chiave di ordinamento.
- Se i dati recenti vengono sottoposti a query più di frequente, specifica la colonna timestamp come colonna principale per la chiave di ordinamento.

  Le query sono più efficaci in quanto possono ignorare interi blocchi che non rientrano nell'intervallo di tempo.
- Se applichi spesso filtri di intervallo o di uguaglianza su una colonna, specifica tale colonna come chiave di ordinamento.

  Amazon Redshift può ignorare la lettura di interi blocchi di dati per quella colonna Può farlo perché tiene traccia dei valori di colonna minimi e massimi archiviati su ogni blocco e può ignorare i blocchi non applicabili all'intervallo di predicati.
- Se si esegue spesso il join di una tabella, specificare la colonna join come chiave di ordinamento e chiave di distribuzione.

  In questo modo, si consente all'ottimizzatore di query di scegliere un sort merge join anziché un hash join più lento. Poiché i dati sono già ordinati nella chiave di join, l'ottimizzatore di query può bypassare la fase di ordinamento del sort merge join.

Scelta del migliore stile di distribuzione

Quando si esegue una query, l'ottimizzatore di query ridistribuisce le righe sui nodi di calcolo per eseguire qualsiasi operazione di join e aggregazione, in base alle necessità. La selezione di uno stile di distribuzione di tabella ha l'obiettivo di minimizzare l'impatto dalle fase di ridistribuzione posizionando i dati dove necessario prima dell'esecuzione della query.

> **Note**
>
> Quando si utilizza l'ottimizzazione automatica della tabella, non è necessario scegliere lo stile di distribuzione della tabella. Per ulteriori informazioni, consultare Utilizzo dell'ottimizzazione automatica delle tabelle.

Seguono alcuni suggerimenti per l'approccio migliore:

1. Distribuire la tabella dei fatti e una tabella di dimensioni sulle relative colonne comuni.

La tabella dei fatti può avere una sola chiave di distribuzione. Le tabelle che eseguono l'operazione di join su un'altra chiave non sono collocate con la tabella dei fatti. Scegliere una dimensione da collocare in base alla frequenza alla quale viene unita in join e alla dimensione delle righe di join. Indicare la chiave primaria della tabella di dimensioni e la chiave esterna corrispondente della tabella dei fatti come DISTKEY.

2. Scegliere la dimensione più grande in base alla dimensione del set di dati filtrato.

   Poiché devono essere distribuite solo le righe utilizzate nel join, considera le dimensioni del set di dati dopo il filtraggio e non le dimensioni della tabella.

3. Scegliere una colonna con un'elevata cardinalità nel set di risultati filtrati.

   Se si distribuisce una tabella delle vendite su una colonna di dati, ad esempio, si otterrebbe probabilmente una distribuzione dei dati abbastanza uniforme, a meno che la maggior parte delle vendite non sia stagionale. Tuttavia, se in genere si utilizza un predicato a intervallo limitato per filtrare su un breve periodo di date, la maggior parte delle righe filtrate si trova su un set di sezioni limitato e il carico di lavoro delle query è asimmetrico.

4. Modificare alcune tabelle di dimensioni per utilizzare la distribuzione ALL.

   Se una tabella di dimensioni non può essere collocata con la tabella dei fatti o altre importanti tabelle di join, è spesso possibile migliorare le prestazioni delle query in modo significativo distribuendo l'intera tabella su tutti i nodi. L'utilizzo della distribuzione ALL moltiplica i requisiti di spazio di storage e aumenta i tempi di caricamento oltre che le operazioni di manutenzione; è quindi necessario valutare tutti i fattori prima di scegliere la distribuzione ALL.

Per permettere ad Amazon Redshift di scegliere lo stile di distribuzione appropriato, specificare `AUTO` per lo stile di distribuzione.

Per ulteriori informazioni sulla scelta degli stili di distribuzione, consultare Utilizzo degli stili di distribuzione dati.

Scelta automatica delle codifiche di compressione con COPY

Puoi specificare le codifiche di compressione al momento della creazione di una tabella, ma nella maggior parte dei casi la compressione automatica produce i migliori risultati.

ENCODE AUTO è l'impostazione di default per le tabelle. Quando la tabella è impostata su ENCODE AUTO, Amazon Redshift gestisce automaticamente la codifica di compressione per tutte le colonne della tabella. Per ulteriori informazioni, consultare CREATE TABLE e ALTER TABLE.

Il comando COPY analizza i dati e applica automaticamente le codifiche di compressione a una tabella vuota nel quadro dell'operazione di caricamento.

La compressione automatica equilibra le prestazioni globali quando si scelgono le codifiche di compressione. Le prestazioni delle scansioni a intervallo limitato potrebbero risultare scadenti se le colonne di chiave di ordinamento vengono compresse più delle altre colonne nella stessa query. Di conseguenza, la compressione automatica sceglie una codifica di compressione meno efficiente per mantenere l'equilibrio tra le colonne di chiave di ordinamento e le altre.

Supponiamo che la chiave di ordinamento della tabelle sia una data o un timestamp e che la tabella utilizzi molte colonne varchar di grandi dimensioni. In questo caso, potresti ottenere delle migliori prestazioni non comprimendo affatto la colonna di chiave di ordinamento. Esegui il comando ANALYZE COMPRESSION sulla tabella, quindi utilizza le codifiche per creare una nuova tabella, ad eccezione della codifica di compressione per la chiave di ordinamento.

La codifica di compressione automatica comporta un costo in termini di prestazioni, ma solo se la tabella è vuota e non ha ancora una codifica di compressione. Per le tabelle di breve durata e per quelle che crei di frequente, come le tabelle di gestione temporanea, carica la tabella una sola volta con la compressione automatica o esegui il comando ANALYZE COMPRESSION e quindi utilizza queste codifiche per creare nuove tabelle. Puoi aggiungere le codifiche all'istruzione CREATE TABLE o utilizzare CREATE TABLE LIKE per creare una nuova tabella con la stessa codifica.

Per ulteriori informazioni, consultare Caricamento di tabelle con compressione automatica.

Definizione dei vincoli di chiave primaria e chiave esterna

Definisci i vincoli di chiave primaria e chiave esterna tra le tabelle quando necessario. Anche se hanno soltanto un valore informativo, l'ottimizzatore di query utilizza tali vincoli per generare piani di query più efficienti.

Non definire vincoli di chiave primaria e di chiave esterna a meno che l'applicazione non applichi i vincoli. Amazon Redshift non applica vincoli di chiave primaria e di chiave esterna univoci.

Per ulteriori informazioni sul modo in cui Amazon Redshift utilizza i vincoli, consultare Definizione di limitazioni delle tabelle.

Utilizzo della dimensione di colonna più piccola possibile

Non utilizzare sistematicamente la dimensione di colonna massima per comodità.

Considera invece i valori più grandi che probabilmente archivierai nelle colonne e dimensionale di conseguenza. Ad esempio, una colonna CHAR per memorizzare le abbreviazioni di stati e territori degli Stati Uniti utilizzate dall'ufficio postale deve essere solo CHAR (2).

Utilizzo di tipi di dati date/time per colonne di dati

Amazon Redshift archivia i dati di tipo DATE e TIMESTAMP più efficacemente rispetto a CHAR o VARCHAR, con conseguente miglioramento delle prestazioni delle query. Utilizza il tipo di dati DATE o TIMESTAMP, in funzione della risoluzione necessaria, anziché un tipo CHAR quando archivi informazioni di tipo date/time. Per ulteriori informazioni, consultare Tipi datetime.

Best practice di Amazon Redshift per il caricamento di dati

Argomenti

- Tutorial sul caricamento dei dati
- Utilizzo di un comando COPY per il caricamento dei dati
- Utilizzo di un singolo comando COPY per il caricamento da più file
- Caricamento di file di dati
- Compressione dei file di dati
- Verifica dei file di dati prima e dopo un caricamento
- Utilizzo di un inserimento multi-riga
- Utilizzo di un inserimento di massa
- Caricamento dei dati nell'ordine della chiave di ordinamento
- Caricamento di dati in blocchi sequenziali
- Utilizzo di tabelle di serie temporali
- Pianificazione per le finestre di manutenzione

Il caricamento di grandi set di dati può richiedere parecchio tempo e consumare una grande quantità di risorse di calcolo. Il modo in cui i dati sono caricati può inoltre avere ripercussioni sulle prestazioni

delle query. Questa sezione presenta le best practice per un caricamento efficace dei dati mediante comandi COPY, inserimenti di massa e tabelle di gestione temporanea.

Tutorial sul caricamento dei dati

Tutorial: Caricamento dei dati da Amazon S3 descrive l'intero processo di caricamento di dati in un bucket Amazon S3 e il processo di caricamento dei dati nelle tabelle mediante il comando COPY. Il tutorial include una guida sulla risoluzione degli errori di caricamento e confronta le differenze nelle prestazioni tra il caricamento da un singolo file e quello da più file.

Utilizzo di un comando COPY per il caricamento dei dati

Il comando COPY carica i dati in parallelo da Amazon S3, Amazon EMR, Amazon DynamoDB o più origini dati su host remoti. Il comando COPY consente di caricare grandi quantità di dati in modo più efficiente rispetto alle istruzioni INSERT e assicura un migliore archiviazione dei dati.

Per ulteriori informazioni sull'utilizzo del comando COPY, consultare Caricamento di dati da Amazon S3 e Caricamento di dati da una tabella Amazon DynamoDB.

Utilizzo di un singolo comando COPY per il caricamento da più file

Amazon Redshift esegue automaticamente il caricamento in parallelo da più file compressi di dati. È possibile specificare i file da caricare tramite il prefisso di un oggetto Amazon S3 o un file manifest.

Tuttavia, se si utilizzano più comandi COPY simultaneamente per caricare una tabella da più file, Amazon Redshift viene forzato a eseguire un caricamento serializzato. Questo tipo di caricamento è molto più lento e richiede un processo VACUUM alla fine se la tabella ha una colonna di ordinamento definita. Per ulteriori informazioni sull'utilizzo del comando COPY per caricare dati in parallelo, consultare Caricamento di dati da Amazon S3.

Caricamento di file di dati

I file di dati di origine sono disponibili in diversi formati e utilizzano diversi algoritmi di compressione. Quando si caricano i dati con il comando COPY, Amazon Redshift carica tutti i file a cui fa riferimento il prefisso del bucket Amazon S3. Un prefisso è una stringa di caratteri all'inizio del nome della chiave dell'oggetto. Se il prefisso si riferisce a più file o file che possono essere divisi, Amazon Redshift carica i dati in parallelo, sfruttando l'architettura MPP di Amazon Redshift. Questo consente di suddividere il carico di lavoro tra i nodi del cluster. Quando tutti i dati vengono caricati da un

singolo file che non è possibile dividere, Amazon Redshift viene forzato a eseguire un caricamento serializzato, che è molto più lento. Le sezioni seguenti descrivono il modo consigliato per caricare diversi tipi di file in Amazon Redshift, a seconda del formato e della compressione.

Caricamento di dati da file che possono essere divisi

I seguenti file possono essere divisi automaticamente quando i dati vengono caricati:

- un file CSV non compresso
- un file CSV compresso con BZIP
- un file a colonne (Parquet/ORC)

Amazon Redshift divide automaticamente i file di almeno 128 MB o più grandi in blocchi. I file a colonne, in particolare Parquet e ORC, non vengono suddivisi se hanno dimensioni inferiori a 128 MB. Redshift utilizza le sezioni che lavorano in parallelo per caricare i dati. Ciò fornisce prestazioni di carico rapide.

Caricamento di dati da file che non possono essere divisi

I tipi di file come JSON o CSV, se compressi con altri algoritmi di compressione, come GZIP, non vengono divisi automaticamente. Per questi si consiglia di dividere manualmente i dati in più file più piccoli di dimensioni simili, da 1 MB a 1 GB dopo la compressione. Fare in modo, inoltre, che il numero di file sia un multiplo del numero di sezioni nel cluster. Per ulteriori informazioni su come suddividere i dati in più file e per esempi di caricamento dei dati con COPY, consulta Caricamento di dati da Amazon S3.

Compressione dei file di dati

Si consiglia di comprimere file di caricamento di grandi dimensioni utilizzando gzip, lzop, bzip2 o Zstandard per comprimerli e dividere i dati in più file di dimensioni minori.

Specifica l'opzione GZIP, LZOP, BZIP2 o ZSTD con il comando COPY. Questo esempio carica la tabella TIME da un file lzop delimitato da pipe.

```
copy time
from 's3://mybucket/data/timerows.lzo'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
lzop
```

```
delimiter '|';
```

Esistono casi in cui non è necessario dividere i file di dati non compressi. Per ulteriori informazioni sulla suddivisione dei dati e per esempi sull'utilizzo di COPY per caricare i dati, consultare Caricamento di dati da Amazon S3.

Verifica dei file di dati prima e dopo un caricamento

Prima di caricare i dati da Amazon S3, verifica prima che il bucket Amazon S3 contenga tutti i file corretti e solo questi. Per ulteriori informazioni, consulta Verifica della presenza dei file corretti nel bucket.

Al termine dell'operazione di caricamento, eseguire una query sulla tabella del sistema STL_LOAD_COMMITS per verificare che i file previsti siano stati caricati. Per ulteriori informazioni, consultare Verifica del caricamento corretto dei dati.

Utilizzo di un inserimento multi-riga

Se un comando COPY non è disponibile e gli inserimenti SQL sono necessari, utilizza un inserimento multi-riga quando possibile. La compressione dei dati risulta inefficiente quando si aggiunge una sola riga o alcune righe alla volta.

Gli inserimenti multi-riga migliorano le prestazioni dividendo in gruppi una serie di inserimenti. L'esempio seguente inserisce tre righe in una tabella con quattro colonne utilizzando una singola istruzione INSERT. Si tratta di un piccolo inserimento, mostrato semplicemente per illustrare la sintassi di un inserimento multi-riga.

```
insert into category_stage values
(default, default, default, default),
(20, default, 'Country', default),
(21, 'Concerts', 'Rock', default);
```

Per maggiori dettagli ed esempi, consultare INSERT.

Utilizzo di un inserimento di massa

Utilizza un'operazione di inserimento di massa con una clausola SELECT per un inserimento di dati ad alte prestazioni.

Utilizza i comandi INSERT e CREATE TABLE AS quando devi spostare dati o un sottoinsieme di dati da una tabella in un'altra.

Ad esempio, la seguente istruzione INSERT seleziona tutte le righe della tabella CATEGORY e le inserisce nella tabella CATEGORY_STAGE.

```
insert into category_stage
(select * from category);
```

L'esempio seguente crea CATEGORY_STAGE come copia di CATEGORY e inserisce tutte le righe di CATEGORY in CATEGORY_STAGE.

```
create table category_stage as
select * from category;
```

Caricamento dei dati nell'ordine della chiave di ordinamento

Carica i dati nell'ordine della chiave di ordinamento per non dover eseguire un vacuum.

Se ogni batch di nuovi dati segue le righe esistenti nella tabella, i data sono archiviati correttamente secondo l'ordinamento previsto e non è necessario eseguire un vacuum. Non è necessario preordinare le righe in ogni caricamento in quanto COPY ordina ogni batch di dati in entrata durante il caricamento.

Ad esempio, supponiamo che carichi dati ogni giorno in base all'attività del giorno corrente. Se la chiave di ordinamento è una colonna timestamp, i dati sono archiviati ordinati. Questo perché i dati del giorno corrente sono sempre aggiunti alla fine dei dati del giorno precedente. Per ulteriori informazioni, consulta Caricamento dei dati nell'ordine delle chiavi di ordinamento. Per ulteriori informazioni sulle operazioni di vacuum, consulta Vacuum delle tabelle.

Caricamento di dati in blocchi sequenziali

Se devi aggiungere una grande quantità di dati, carica i dati in blocchi sequenziali in base al tipo di ordinamento per eliminare la necessità di eseguire un vacuum.

Ad esempio, supponiamo che devi caricare una tabella con eventi da gennaio 2017 a dicembre 2017. Supponendo che ogni mese si trovi in un singolo file, carica le righe per gennaio, poi per febbraio e così via. La tabella è completamente ordinata al termine del caricamento e non è necessario eseguire un vacuum. Per ulteriori informazioni, consultare Utilizzo di tabelle di serie temporali.

Quando si caricano set di dati di grandi dimensioni, è possibile che lo spazio necessario per l'ordinamento sia superiore a quello disponibile. Caricando i dati in blocchi più piccoli, utilizzi molto meno spazio di ordinamento intermedio durante ogni caricamento. Inoltre, il caricamento di blocchi più piccoli facilita il riavvio se COPY non riesce e viene annullato.

Utilizzo di tabelle di serie temporali

Se i tuoi file hanno un periodo di conservazione fisso, puoi organizzare i dati come una sequenza di tabelle di serie temporali. In questa sequenza, ogni tabella è identica all'altra ma contiene dati per differenti intervalli di tempo.

Puoi facilmente eliminare i dati obsoleti eseguendo semplicemente un comando DROP TABLE sulle tabelle corrispondenti. Questo approccio è molto più rapido dell'esecuzione di un comando DELETE su vasta scala ed evita la necessità di eseguire un VACUUM per recuperare spazio. Puoi creare una vista UNION ALL per nascondere il fatto che i dati siano archiviati in tabelle differenti. Quando elimini dati obsoleti, perfeziona la vista UNION ALL per rimuovere le tabelle eliminate. Analogamente, quando carichi nuovi periodi di tempo in nuove tabelle, aggiungi nuove tabelle alla vista. Per segnalare all'ottimizzatore di saltare la scansione sulle tabelle che non corrispondono al filtro di query, la definizione della vista filtra in base all'intervallo di date che corrisponde a ciascuna tabella.

Evita di avere troppe tabelle nella vista UNION ALL. Ogni ulteriore tabella aggiunge un tempo di elaborazione alla query. Non è necessario che le tabelle usino lo stesso intervallo di tempo. Ad esempio, potresti avere tabelle per periodi di tempo diversi (giornaliero, mensile e annuale).

Se utilizzi tabelle di serie temporali con una colonna timestamp per la chiave di ordinamento, carichi i dati nell'ordine della chiave di ordinamento. In questo modo, si elimina la necessità di eseguire un vacuum per riordinare i dati. Per ulteriori informazioni, consulta Caricamento dei dati nell'ordine delle chiavi di ordinamento.

Pianificazione per le finestre di manutenzione

Se durante l'esecuzione di una query viene avviata una manutenzione pianificata, la query viene terminata e annullata e dovrà quindi essere eseguita nuovamente. Pianifica le operazioni di lunga durata, come i caricamenti di grandi quantità di dati o un'operazione VACUUM, in modo da evitare le finestre di manutenzione. Puoi anche ridurre al minimo tale rischio e facilitare i riavvii quando questi sono necessari eseguendo i caricamenti di dati in incrementi più piccoli e gestendo la dimensione delle operazioni VACUUM. Per ulteriori informazioni, consultare Caricamento di dati in blocchi sequenziali e Vacuum delle tabelle.

Best practice di Amazon Redshift per la progettazione di query

Per ottimizzare le prestazioni delle query, segui queste raccomandazioni durante la creazione di query.

- Progetta le tabelle conformemente alle best practice in modo da fornire basi solide per le prestazioni delle query. Per ulteriori informazioni, consultare Best practice di Amazon Redshift per la progettazione di tabelle.
- Non utilizzare `select *`. Includi solo le colonne di cui hai bisogno.
- Utilizza Espressione condizionale CASE per eseguire aggregazioni complesse anziché selezionare più volte dalla stessa tabella.
- Non utilizzare cross join se non assolutamente necessario. Questi join senza una condizione di join generano un prodotto cartesiano di due tabelle. I cross join sono in genere eseguiti come nested loop join, ovvero i tipi di join più lenti.
- Usa le subquery nei casi in cui una tabella nella query è utilizzata solo per le condizioni di predicato e la subquery restituisce un numero esiguo di righe (inferiore a 200). L'esempio seguente utilizza una subquery per evitare di unire in join la tabella LISTING.

  ```
  select sum(sales.qtysold)
  from sales
  where salesid in (select listid from listing where listtime > '2008-12-26');
  ```

- Utilizza predicati per limitare il più possibile il set di dati.
- Nel predicato, utilizza gli operatori meno onerosi. Gli operatori Condizione di confronto sono preferibili agli operatori LIKE. Gli operatori LIKE sono ancora preferibili a SIMILAR TO o Operatori POSIX.
- Evita l'utilizzo di funzioni nei predicati delle query. Il loro utilizzo può accrescere il costo della query in quanto richiedono un numero elevato di righe per risolvere le fasi intermedie della query.
- Se possibile, utilizza una clausola WHERE per limitare il set di dati. In tal modo, il pianificatore di query può utilizzare l'ordine delle righe per determinare quali record corrispondono ai criteri e quindi ignorare la scansione di grandi quantità di blocchi del disco. In caso contrario, il motore di esecuzione di query deve eseguire la scansione della totalità delle colonne partecipanti.
- Aggiungi predicati per filtrare le tabelle che partecipano alle operazioni di join, anche se i predicati applicano gli stessi filtri. La query restituisce lo stesso set di risultati, ma Amazon Redshift è in grado di filtrare le tabelle join prima della fase di scansione e può quindi ignorare in modo efficace

la scansione dei blocchi di quelle tabelle. I filtri ridondanti non sono necessari se filtri una colonna utilizzata nella condizione di join.

Ad esempio, supponiamo che intendi eseguire un join di SALES con LISTING per trovare le vendite di biglietti per i biglietti elencati dopo dicembre, raggruppati per venditore. Entrambe le tabelle sono ordinate per data. La query seguente unisce in join le tabelle sulla relativa chiave comune e filtra i valori `listing.listtime` posteriori al 1° dicembre.

```
select listing.sellerid, sum(sales.qtysold)
from sales, listing
where sales.salesid = listing.listid
and listing.listtime > '2008-12-01'
group by 1 order by 1;
```

La clausola WHERE non include un predicato for `sales.saletime`, quindi il motore di esecuzione è costretto a eseguire la scansione dell'intera tabella SALES. Se sai che il filtro comporterà un minor numero di righe partecipanti al join, aggiungi anche quel filtro. L'esempio seguente riduce il tempo di esecuzione in modo significativo.

```
select listing.sellerid, sum(sales.qtysold)
from sales, listing
where sales.salesid = listing.listid
and listing.listtime > '2008-12-01'
and sales.saletime > '2008-12-01'
group by 1 order by 1;
```

- Utilizza chiavi di ordinamento nella clausola GROUP BY di modo che il pianificatore di query possa utilizzare un'aggregazione più efficace. Una query può essere inclusa in un'aggregazione in una fase quando il relativo elenco GROUP BY contiene solo colonne di chiave di ordinamento, una delle quali è anche la chiave di distribuzione. Le colonne di chiave di ordinamento nell'elenco GROUP BY devono includere la prima chiave di ordinamento e quindi altre chiavi di ordinamento che intendi utilizzare. Ad esempio, è consentito utilizzare la prima chiave di ordinamento, la prima e la seconda chiave di ordinamento, la prima, la seconda e la terza chiave di ordinamento e così via. Non è possibile utilizzare la prima e la terza chiave di ordinamento.

  Puoi confermare l'utilizzo di un'aggregazione a una fase eseguendo il comando EXPLAIN e cercando `XN GroupAggregate` nella fase di aggregazione della query.
- Se utilizzi le clausole GROUP BY e ORDER BY, assicurati di posizionare le colonne in entrambe nello stesso ordine. In altre parole, utilizza l'approccio seguente.

```
group by a, b, c
order by a, b, c
```

Non utilizzare l'approccio seguente.

```
group by b, c, a
order by a, b, c
```

Utilizzo dei suggerimenti da Amazon Redshift Advisor

Per migliorare le prestazioni e ridurre i costi operativi del cluster Amazon Redshift, Amazon Redshift Advisor fornisce suggerimenti specifici sulle modifiche da implementare. Advisor sviluppa raccomandazioni personalizzate analizzando le prestazioni e i parametri di utilizzo relativi al cluster. Queste raccomandazioni riguardano le impostazioni delle operazioni e del cluster. Per aiutarti a stabilire le priorità nelle tue ottimizzazioni, Advisor classifica le raccomandazioni per ordine di impatto.

Advisor basa le sue raccomandazioni sull'osservazione delle statistiche delle prestazioni o dei dati delle operazioni. Advisor sviluppa le osservazioni eseguendo test sui cluster per determinare se un valore di test rientra nell'intervallo specificato. Se il risultato del test non rientra in quell'intervallo, Advisor genera un'osservazione per quel cluster. Nello stesso tempo, crea una raccomandazione su come riportare il valore osservato nell'intervallo della best practice. Advisor visualizza soltanto le raccomandazioni che dovrebbero avere un impatto significativo sulle prestazioni e sulle operazioni. Quando Advisor determina che una raccomandazione è stata applicata, la rimuove dall'elenco delle raccomandazioni.

Supponiamo, ad esempio, che il tuo data warehouse contenga un numero elevato di colonne di tabella non compresse. In questo caso, puoi ridurre i costi di storage del cluster ricostruendo le tabelle mediante il parametro `ENCODE` per specificare la compressione delle colonne. Adesso, supponiamo invece che Advisor osservi che il cluster contiene un numero elevato di dati di tabella non compressi. In questo caso, fornisce il blocco di codice SQL per trovare le colonne di tabella suscettibili di essere compresse e le risorse che descrivono come comprimere tali colonne.

Regioni di Amazon Redshift

La funzionalità Amazon Redshift Advisor è disponibile solo nelle seguenti regioni: AWS

- Regione Stati Uniti orientali (Virginia settentrionale) (us-east-1)

- Regione Stati Uniti orientali (Ohio) (us-east-2)
- Regione Stati Uniti occidentali (California settentrionale) (us-west-1)
- Regione Stati Uniti occidentali (Oregon) (us-west-2)
- Regione Africa (Città del Capo) (af-south-1)
- Regione Asia Pacifico (Hong Kong) (ap-east-1)
- Regione Asia Pacifico (Hyderabad) (ap-south-2)
- Regione Asia Pacifico (Jakarta) (ap-southeast-3)
- Regione Asia Pacifico (Melbourne) (ap-southeast-4)
- Regione Asia Pacifico (Mumbai) (ap-south-1)
- Regione Asia Pacifico (Osaka) (ap-northeast-3)
- Regione Asia Pacifico (Seoul) (ap-northeast-2)
- Regione Asia Pacifico (Singapore) (ap-southeast-1)
- Regione Asia Pacifico (Sydney) (ap-southeast-2)
- Regione Asia Pacifico (Tokyo) (ap-northeast-1)
- Regione Canada (Centrale) (ca-central-1)
- Regione Canada occidentale (Calgary) (ca-west-1)
- Regione Cina (Pechino) (cn-north-1)
- Regione Cina (Ningxia) (cn-northwest-1)
- Regione Europa (Francoforte) (eu-central-1)
- Regione Europa (Irlanda) (eu-west-1)
- Regione Europa (Londra) (eu-west-2)
- Regione Europa (Milano) (eu-south-1)
- Regione Europa (Parigi) (eu-west-3)
- Regione Europa (Spagna) (eu-south-2)
- Regione Europa (Stoccolma) (eu-north-1)
- Regione Europa (Zurigo) (eu-central-2)
- Regione di Israele (Tel Aviv) (il-central-1)
- Regione Medio Oriente (Bahrein) (me-south-1)
- Regione Medio Oriente (EAU) (me-central-1)
- Regione Sud America (San Paolo) (sa-east-1)

Argomenti

Visualizzazione dei consigli di Amazon Redshift Advisor

Puoi accedere ai consigli di Amazon Redshift Advisor utilizzando la console Amazon Redshift, l'API Amazon Redshift o. AWS CLI Per accedere ai consigli devi avere l'autorizzazione `redshift:ListRecommendations` associata al tuo ruolo o identità IAM.

Visualizzazione dei consigli di Amazon Redshift Advisor sulla console con provisioning di Amazon Redshift

Puoi visualizzare i consigli di Amazon Redshift Advisor su. AWS Management Console

Per visualizzare i consigli di Amazon Redshift Advisor per i cluster Amazon Redshift sulla console

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Advisor.
3. Espandere ogni raccomandazione per visualizzare ulteriori dettagli. In questa pagina è possibile ordinare e raggruppare le raccomandazioni.

Visualizzazione dei consigli di Amazon Redshift Advisor utilizzando le operazioni dell'API Amazon Redshift

Puoi elencare i consigli di Amazon Redshift Advisor per i cluster Amazon Redshift utilizzando l'API Amazon Redshift. In genere, sviluppi e applichi nel linguaggio di programmazione che preferisci per chiamare l'`redshift:ListRecommendations`API utilizzando un SDK. AWS Per ulteriori informazioni, ListRecommendationsconsulta Amazon Redshift API Reference.

Visualizzazione dei consigli di Amazon Redshift Advisor utilizzando le operazioni AWS Command Line Interface

Puoi elencare i consigli di Amazon Redshift Advisor per i cluster Amazon Redshift utilizzando il. AWS Command Line InterfacePer ulteriori informazioni, consulta list-recommendations nel Command Reference.AWS CLI

Suggerimenti di Amazon Redshift Advisor

Amazon Redshift Advisor fornisce raccomandazioni su come ottimizzare il cluster Amazon Redshift per migliorare le prestazioni e ridurre i costi operativi. Puoi trovare delle spiegazioni su ogni raccomandazione nella console, come descritto precedentemente. Ulteriori dettagli sulle raccomandazioni sono forniti nelle sezioni seguenti.

Argomenti

Compressione degli oggetti file di Amazon S3 caricati mediante COPY

Il comando COPY sfrutta i vantaggi dell'architettura MPP (Massively Parallel Processing) di Amazon Redshift per leggere e caricare dati in parallelo. Può leggere file da Amazon S3, tabelle DynamoDB e un output di testo da uno o più host remoti.

In caso di caricamento di grandi quantità di dati, ti consigliamo vivamente di utilizzare il comando COPY per caricare file di dati compressi da S3. La compressione di set di dati di grandi dimensioni consente di risparmiare tempo durante il caricamento dei file in Amazon S3. COPY può anche accelerare il processo di caricamento decomprimendo i file man mano che vengono letti.

Analisi

Le prestazioni dei comandi COPY di lunga durata che caricano set di dati voluminosi non compressi spesso possono essere migliorate considerevolmente. L'analisi di Advisor identifica i comandi COPY

che caricano set di dati voluminosi non compressi. In tal caso, Advisor genera un suggerimento per implementare la compressione sui file di origine in Amazon S3.

Raccomandazione

Accertati che ogni comando COPY che carica una quantità importante di dati o che viene eseguito per un lungo periodo di tempo, importi oggetti dati compressi da Amazon S3. Puoi identificare i comandi COPY che caricano set di dati non compressi voluminosi da Amazon S3 eseguendo il comando SQL seguente come utente con privilegi avanzati.

```
SELECT
    wq.userid, query, exec_start_time AS starttime, COUNT(*) num_files,
    ROUND(MAX(wq.total_exec_time/1000000.0),2) execution_secs,
    ROUND(SUM(transfer_size)/(1024.0*1024.0),2) total_mb,
    SUBSTRING(querytxt,1,60) copy_sql
FROM stl_s3client s
JOIN stl_query q USING (query)
JOIN stl_wlm_query wq USING (query)
WHERE s.userid>1 AND http_method = 'GET'
    AND POSITION('COPY ANALYZE' IN querytxt) = 0
    AND aborted = 0 AND final_state='Completed'
GROUP BY 1, 2, 3, 7
HAVING SUM(transfer_size) = SUM(data_size)
AND SUM(transfer_size)/(1024*1024) >= 5
ORDER BY 6 DESC, 5 DESC;
```

Se i dati in gestione temporanea rimangono in Amazon S3 dopo il caricamento, evento comune nelle architetture dei data lake, l'archiviazione di questi dati in un formato compresso può ridurre i costi di archiviazione.

Consigli di implementazione

- La dimensione ideale dell'oggetto è di 1-128 MB dopo la compressione.
- Puoi comprimere i file con il formato gzip, lzop o bzip2.

Isolare molteplici database attivi

Come best practice, consigliamo di isolare i database in Amazon Redshift gli uni dagli altri. Le query sono eseguite in uno specifico database e non possono accedere ai dati da qualsiasi altro database

sul cluster. Tuttavia, le query eseguite in tutti i database di un cluster condividono lo stesso spazio di storage di cluster sottostante e le stesse risorse di calcolo. Quando un singolo cluster contiene molteplici database attivi, i relativi carichi di lavoro sono spesso non correlati.

Analisi

L'analisi di Advisor esamina tutti i database sul cluster per verificare la presenza di carichi di lavoro attivi in esecuzione nello stesso momento. Se questi carichi sono presenti, Advisor genera un suggerimento per prendere in considerazione la migrazione dei database verso cluster Amazon Redshift distinti.

Raccomandazione

Prendi in considerazione lo spostamento di ogni database sottoposto a query a un cluster dedicato distinto. Utilizzando un cluster distinto è possibile ridurre i conflitti tra le risorse e migliorare le prestazioni delle query. Questo perché è possibile definire la dimensione di ogni cluster in funzione delle esigenze di archiviazione, costi e prestazioni di ogni carico di lavoro. Inoltre, i carichi di lavoro non correlati spesso utilizzano differenti configurazioni di gestione del carico di lavoro.

Per identificare quali database sono attivamente utilizzati, puoi eseguire questo comando SQL come utente con privilegi avanzati.

```
SELECT database,
  COUNT(*) as num_queries,
  AVG(DATEDIFF(sec,starttime,endtime)) avg_duration,
  MIN(starttime) as oldest_ts,
  MAX(endtime) as latest_ts
FROM stl_query
WHERE userid > 1
GROUP BY database;
```

Consigli di implementazione

- Poiché un utente deve connettersi a ogni database in modo specifico e che le query possono accedere solo a un singolo database, lo spostamento dei database in cluster distinti ha un impatto minimo per gli utenti.
- Per spostare un database, procedere come segue:
  1. Ripristinare temporaneamente uno snapshot del cluster corrente a un cluster della stessa dimensione.

2. Eliminare tutti i database dal nuovo cluster, ad eccezione del database target da spostare.
3. Ridimensionare il cluster su un tipo di nodo appropriato e prendere in considerazione il carico di lavoro del database.

Riallocazione della memoria WLM

Amazon Redshift instrada le query degli utenti a Implementazione di WLM manuale per l'elaborazione. Le modalità con cui le query vengono instradate alle code sono definite dalla gestione del carico di lavoro (WLM). Amazon Redshift assegna a ogni coda una parte della memoria disponibile del cluster. La memoria di una coda viene ripartita tra gli slot di query della coda.

Quando una coda è configurata con più slot di quelli richiesti dal carico di lavoro, la memoria allocata a questi slot inutilizzati è sottoutilizzata. La riduzione degli slot configurati in funzione delle esigenze del carico di lavoro massimo consente di ridistribuire la memoria sottoutilizzata agli slot attivi e può quindi migliorare le prestazioni delle query.

Analisi

L'analisi di Advisor esamina le esigenze in materia di simultaneità del carico di lavoro per identificare le code di query con slot non utilizzati. Advisor genera una raccomandazione per ridurre il numero di slot in una coda quando trova quanto segue:

- Una coda con slot che sono completamente inattivi durante l'analisi.
- Una coda con più di quattro slot di cui almeno due sono inattivi durante l'analisi.

Raccomandazione

La riduzione degli slot configurati in funzione delle esigenze del carico di lavoro massimo consente di ridistribuire la memoria sottoutilizzata agli slot attivi. Prendi in considerazione la riduzione del numero di slot configurati per le code i cui slot non sono mai stati completamente utilizzati. Per identificare queste code, puoi comparare le esigenze orarie massime degli slot di ogni coda eseguendo il comando SQL seguente come utente con privilegi avanzati.

```
WITH
 generate_dt_series AS (select sysdate - (n * interval '5 second') as dt from (select
 row_number() over () as n from stl_scan limit 17280)),
 apex AS (
```

```
    SELECT iq.dt, iq.service_class, iq.num_query_tasks, count(iq.slot_count) as
 service_class_queries, sum(iq.slot_count) as service_class_slots
    FROM
        (select gds.dt, wq.service_class, wscc.num_query_tasks, wq.slot_count
        FROM stl_wlm_query wq
        JOIN stv_wlm_service_class_config wscc ON (wscc.service_class =
 wq.service_class AND wscc.service_class > 5)
        JOIN generate_dt_series gds ON (wq.service_class_start_time <= gds.dt AND
 wq.service_class_end_time > gds.dt)
        WHERE wq.userid > 1 AND wq.service_class > 5) iq
    GROUP BY iq.dt, iq.service_class, iq.num_query_tasks),
    maxes as (SELECT apex.service_class, trunc(apex.dt) as d, date_part(h,apex.dt) as
 dt_h, max(service_class_slots) max_service_class_slots
                    from apex group by apex.service_class, apex.dt,
 date_part(h,apex.dt))
SELECT apex.service_class - 5 AS queue, apex.service_class, apex.num_query_tasks AS
 max_wlm_concurrency, maxes.d AS day, maxes.dt_h || ':00 - ' || maxes.dt_h || ':59' as
 hour, MAX(apex.service_class_slots) as max_service_class_slots
FROM apex
JOIN maxes ON (apex.service_class = maxes.service_class AND apex.service_class_slots =
 maxes.max_service_class_slots)
GROUP BY  apex.service_class, apex.num_query_tasks, maxes.d, maxes.dt_h
ORDER BY apex.service_class, maxes.d, maxes.dt_h;
```

La colonna `max_service_class_slots` rappresenta il numero massimo di slot di query WLM nella coda di query per quell'ora. Se esistono delle code sottoutilizzate, implementare l'ottimizzazione della riduzione degli slot modificando un gruppo di parametri, come descritto nella Guida alla gestione di Amazon Redshift.

Consigli di implementazione

- Se il tuo carico di lavoro è fortemente variabile in volume, assicurati che l'analisi catturi un periodo di utilizzazione massima. In caso contrario, esegui il comando SQL precedente ripetutamente per monitorare le esigenze massime in materia di simultaneità.
- Per ulteriori dettagli sull'interpretazione dei risultati delle query dal codice SQL precedente, vedete lo script wlm_apex_hourly.sql su. GitHub

Ignorare l'analisi di compressione durante l'esecuzione del comando COPY

Quando i dati vengono caricati in una tabella vuota con codifica di compressione dichiarata con il comando COPY, Amazon Redshift applica la compressione dell'archiviazione. Questa ottimizzazione assicura che i dati nel cluster siano archiviati in modo efficace anche quando caricati da utenti finali. L'analisi richiesta per applicare la compressione può durare un certo tempo.

Analisi

L'analisi di Advisor cerca le operazioni COPY che sono state ritardate dall'analisi di compressione automatica. L'analisi determina le codifiche di compressione campionando i dati durante il caricamento degli stessi. Questo campionamento è simile a quello eseguito dal comando ANALYZE COMPRESSION.

Quando carichi dei dati nel quadro di un processo strutturato, come un batch ETL durante la notte, puoi definire la compressione in anticipo. Puoi anche ottimizzare le definizioni di tabella per ignorare in modo permanente questa fase senza alcun impatto negativo.

Raccomandazione

Per migliorare la reattività del comando COPY omettendo la fase di analisi della compressione, eseguire una delle due seguenti operazioni:

- Utilizza il parametro di colonna `ENCODE` durante la creazione delle tabelle che carichi mediante il comando COPY.
- Disattiva totalmente la compressione indicando il parametro `COMPUPDATE OFF` nel comando COPY.

La migliore soluzione è generalmente di utilizzare la codifica di colonna durante la creazione di tabelle in quanto questo approccio consente di archiviare dati compressi sul disco. Puoi utilizzare il comando ANALYZE COMPRESSION per suggerire le codifiche di compressione, ma devi ricreare la tabella per applicare queste codifiche. Per automatizzare questo processo, è possibile utilizzare l' AWSColumnEncodingutilità, disponibile su. GitHub

Per identificare le operazioni COPY recenti che hanno generato l'analisi di compressione automatica, esegui il comando SQL seguente.

```
WITH xids AS (
```

```
    SELECT xid FROM stl_query WHERE userid>1 AND aborted=0
    AND querytxt = 'analyze compression phase 1' GROUP BY xid
    INTERSECT SELECT xid FROM stl_commit_stats WHERE node=-1)
SELECT a.userid, a.query, a.xid, a.starttime, b.complyze_sec,
    a.copy_sec, a.copy_sql
FROM (SELECT q.userid, q.query, q.xid, date_trunc('s',q.starttime)
    starttime, substring(querytxt,1,100) as copy_sql,
    ROUND(datediff(ms,starttime,endtime)::numeric / 1000.0, 2) copy_sec
    FROM stl_query q JOIN xids USING (xid)
    WHERE (querytxt ilike 'copy %from%' OR querytxt ilike '% copy %from%')
    AND querytxt not like 'COPY ANALYZE %') a
LEFT JOIN (SELECT xid,
    ROUND(sum(datediff(ms,starttime,endtime))::numeric / 1000.0,2) complyze_sec
    FROM stl_query q JOIN xids USING (xid)
    WHERE (querytxt like 'COPY ANALYZE %'
    OR querytxt like 'analyze compression phase %')
    GROUP BY xid ) b ON a.xid = b.xid
WHERE b.complyze_sec IS NOT NULL ORDER BY a.copy_sql, a.starttime;
```

Consigli di implementazione

- Assicurati che tutte le tabelle di dimensione significativa create durante i processi ETL (ad esempio, le tabelle di gestione temporanea e le tabelle temporanee) dichiarino una codifica di compressione per tutte le colonne ad eccezione della prima chiave di ordinamento.
- Stima la dimensione della tabella in corso di caricamento per tutta la sua durata di vita per ogni comando COPY identificato dal comando SQL precedente. Se ritieni che la tabella resterà estremamente piccola, disattiva completamente la compressione con il parametro `COMPUPDATE OFF`. In caso contrario, crea la tabella con una compressione esplicita prima di caricarla con il comando COPY.

Suddivisione degli oggetti Amazon S3 caricati da COPY

Il comando COPY sfrutta i vantaggi dell'architettura MPP (Massively Parallel Processing) di Amazon Redshift per leggere e caricare dati da file su Amazon S3. Il comando COPY consente di caricare dati in parallelo da più file, suddividendo il carico di lavoro tra i nodi del cluster. Per ottenere un throughput ottimale, ti consigliamo vivamente di suddividere i dati in più file per beneficiare dall'elaborazione parallela.

Analisi

L'analisi di Advisor identifica i comandi COPY che caricano grandi set di dati contenuti in un piccolo numero di file gestiti temporaneamente in Amazon S3. Le prestazioni dei comandi COPY di lunga durata che caricano set di dati voluminosi da alcuni file possono spesso essere migliorate considerevolmente. Quando Advisor rileva che l'esecuzione di questi comandi COPY richiede parecchio tempo, crea un suggerimento per aumentare il parallelismo suddividendo i dati in ulteriori file in Amazon S3.

Raccomandazione

In questo caso, consigliamo le seguenti operazioni, elencate in ordine di priorità:

1. Ottimizzazione dei comandi COPY che caricano un numero di file inferiore al numero di nodi cluster.
2. Ottimizzazione dei comandi COPY che caricano un numero di file inferiore al numero di sezioni di cluster.
3. Ottimizzazione dei comandi COPY dove il numero di file non è un multiplo del numero di sezioni nel cluster.

Determinati comandi COPY caricano una notevole quantità di dati o hanno un tempo di esecuzione piuttosto lungo. Per questi comandi consigliamo di caricare un numero di oggetti dati da Amazon S3 equivalente a un multiplo del numero di sezioni nel cluster. Per identificare il numero di oggetti S3 che ogni comando COPY ha caricato, eseguire il codice SQL seguente come utente con privilegi avanzati.

```
SELECT
    query, COUNT(*) num_files,
    ROUND(MAX(wq.total_exec_time/1000000.0),2) execution_secs,
    ROUND(SUM(transfer_size)/(1024.0*1024.0),2) total_mb,
    SUBSTRING(querytxt,1,60) copy_sql
FROM stl_s3client s
JOIN stl_query q USING (query)
JOIN stl_wlm_query wq USING (query)
WHERE s.userid>1 AND http_method = 'GET'
    AND POSITION('COPY ANALYZE' IN querytxt) = 0
    AND aborted = 0 AND final_state='Completed'
GROUP BY query, querytxt
HAVING (SUM(transfer_size)/(1024*1024))/COUNT(*) >= 2
ORDER BY CASE
WHEN COUNT(*) < (SELECT max(node)+1 FROM stv_slices) THEN 1
```

```
WHEN COUNT(*) < (SELECT COUNT(*) FROM stv_slices WHERE node=0) THEN 2
ELSE 2+((COUNT(*) % (SELECT COUNT(*) FROM stv_slices))/(SELECT COUNT(*)::DECIMAL FROM
 stv_slices))
END, (SUM(transfer_size)/(1024.0*1024.0))/COUNT(*) DESC;
```

Consigli di implementazione

- Il numero di sezioni in un nodo dipende dalla dimensione dei nodi del cluster. Per ulteriori informazioni sul numero di sezioni nei vari tipi di nodo, consulta Cluster e nodi in Amazon Redshift nella Guida alla gestione di Amazon Redshift.
- È possibile caricare più file specificando un prefisso comune o chiave di prefisso, per l'insieme o elencando in modo esplicito i file in un file manifest. Per ulteriori informazioni sul caricamento di file, consultare Caricamento dei dati da file compressi e non compressi.
- Amazon Redshift non prende in considerazione la dimensione di file nella suddivisione del carico di lavoro. Suddividi i file di dati di caricamento di modo che siano all'incirca della stessa dimensione, tra 1 MB e 1 GB dopo la compressione.

Aggiornamento delle statistiche delle tabelle

Amazon Redshift utilizza un ottimizzatore di query basato sui costi per scegliere il piano di esecuzione ottimale per le query. Le stime dei costi sono basate sulle statistiche delle tabelle raccolte utilizzando il comando ANALYZE. Quando le statistiche di una tabella risultano non aggiornate o mancanti, il database potrebbe scegliere un piano meno efficiente per l'esecuzione delle query, soprattutto per quelle complesse. Disporre di statistiche aggiornate consente di eseguire query complesse nel minor tempo possibile.

Analisi

L'analisi Advisor tiene traccia delle tabelle le cui statistiche sono out-of-date o mancano. Esamina i metadati di accesso alle tabelle che sono associati a query complesse. Se nelle tabelle a cui si accede di frequente con modelli complessi mancano delle statistiche, Advisor crea una raccomandazione critica per eseguire ANALYZE. Se le tabelle a cui si accede frequentemente con modelli complessi contengono out-of-date statistiche, Advisor suggerisce di eseguire ANALYZE.

Raccomandazione

Ogni volta che il contenuto delle tabelle cambia, aggiorna le statiche con ANALYZE. Consigliamo di eseguire ANALYZE ogni volta che un numero importante di nuove righe di dati sono caricate in una tabella esistente con il comando COPY o INSERT. Inoltre, consigliamo di eseguire ANALYZE ogni volta che un numero importante di nuove righe vengono modificate in una tabella esistente con il comando UPDATE o DELETE. Per identificare le tabelle mancanti o out-of-date statistiche, esegui il seguente comando SQL come superutente. I risultati sono ordinati dalla tabella più grande a quella più piccola.

Per identificare le tabelle mancanti o out-of-date le statistiche, esegui il seguente comando SQL come superutente. I risultati sono ordinati dalla tabella più grande a quella più piccola.

```
SELECT
   ti.schema||'.'||ti."table" tablename,
   ti.size table_size_mb,
   ti.stats_off statistics_accuracy
 FROM svv_table_info ti
 WHERE ti.stats_off > 5.00
 ORDER BY ti.size DESC;
```

Consigli di implementazione

Per impostazione predefinita, la soglia di ANALYZE è impostata su 10 percento. Questa impostazione predefinita significa che il comando ANALYZE salta una determinata tabella se la percentuale di righe che sono state modificate dall'ultimo ANALYZE è inferiore al 10 percento. Di conseguenza, potresti scegliere di generare comandi ANALYZE alla fine di ogni processo ETL. Con questo approccio, ANALYZE viene spesso ignorato ma se ne assicura l'esecuzione quando necessario.

Le statistiche di ANALYZE hanno un impatto maggiore sulle colonne utilizzate nei join (ad esempio, `JOIN tbl_a ON col_b`) o come predicati (ad esempio, `WHERE col_b = 'xyz'`). Per impostazione predefinita, ANALYZE raccoglie statistiche per tutte le colonne nella tabella specificata. Se necessario, puoi ridurre il tempo di esecuzione di ANALYZE eseguendo questo comando solo per le colonne su cui ha il maggiore impatto. Puoi utilizzare i comandi SQL seguenti per identificare le colonne utilizzate come predicati. È possibile anche lasciare a Amazon Redshift la scelta delle colonne da analizzare specificando `ANALYZE PREDICATE COLUMNS`.

```
WITH predicate_column_info as (
```

```
SELECT ns.nspname AS schema_name, c.relname AS table_name, a.attnum as col_num,
 a.attname as col_name,
        CASE
            WHEN 10002 = s.stakind1 THEN array_to_string(stavalues1, '||')
            WHEN 10002 = s.stakind2 THEN array_to_string(stavalues2, '||')
            WHEN 10002 = s.stakind3 THEN array_to_string(stavalues3, '||')
            WHEN 10002 = s.stakind4 THEN array_to_string(stavalues4, '||')
            ELSE NULL::varchar
        END AS pred_ts
   FROM pg_statistic s
   JOIN pg_class c ON c.oid = s.starelid
   JOIN pg_namespace ns ON c.relnamespace = ns.oid
   JOIN pg_attribute a ON c.oid = a.attrelid AND a.attnum = s.staattnum)
SELECT schema_name, table_name, col_num, col_name,
       pred_ts NOT LIKE '2000-01-01%' AS is_predicate,
       CASE WHEN pred_ts NOT LIKE '2000-01-01%' THEN (split_part(pred_ts,
 '||',1))::timestamp ELSE NULL::timestamp END as first_predicate_use,
       CASE WHEN pred_ts NOT LIKE '%||2000-01-01%' THEN (split_part(pred_ts,
 '||',2))::timestamp ELSE NULL::timestamp END as last_analyze
FROM predicate_column_info;
```

Per ulteriori informazioni, consultare Analisi delle tabelle.

Abilitazione dell'accelerazione di query brevi

L'accelerazione di query brevi (SQA) rende prioritarie le query a esecuzione breve rispetto a quelle a esecuzione prolungata. L'accelerazione di query brevi (SQA, Short Query Acceleration) esegue query a esecuzione breve in uno spazio dedicato, in modo che le query SQA non siano costrette ad attendere in coda dietro query più lunghe. SQA assegna la priorità solo alle query che hanno un'esecuzione breve e si trovano in una coda definita dall'utente. Con SQA, le query a esecuzione breve iniziano l'esecuzione più rapidamente e gli utenti visualizzano i risultati prima.

Se abiliti SQA, puoi ridurre o eliminare le code di gestione del carico di lavoro (WLM) dedicate all'esecuzione di query brevi. Inoltre, le query a esecuzione prolungata non devono contendere gli slot con le query brevi in una coda, quindi puoi configurare le code WLM per utilizzare un numero inferiore di slot di query. Quando usi una simultaneità inferiore, il throughput delle query aumenta e le prestazioni generali del sistema risultano migliorate per la maggior parte dei carichi di lavoro. Per ulteriori informazioni, consultare Utilizzo dall'accelerazione di query brevi.

Analisi

Advisor controlla i modelli del carico di lavoro e segnala il numero di query recenti in cui SQA ridurrebbe la latenza e il tempo di coda giornaliero per le query idonee per SQA.

Raccomandazione

Modifica della configurazione WLM per attivare SQA. Amazon Redshift utilizza un algoritmo di machine learning per analizzare ciascuna query idonea. Le previsioni migliorano man mano che SQA assimila i tuoi modelli di query. Per ulteriori informazioni, consultare Configurazione della gestione del carico di lavoro.

Quando attivi SQA, WLM imposta il tempo di esecuzione massimo per le query brevi su dinamico per impostazione predefinita. Ti consigliamo di mantenere le impostazioni dinamiche per il tempo massimo di esecuzione SQA.

Consigli di implementazione

Per controllare se SQA è attivato, esegui la seguente query. Se la query restituisce una riga, SQA è attivato.

```
select * from stv_wlm_service_class_config
where service_class = 14;
```

Per ulteriori informazioni, consulta Monitoraggio dell'accelerazione di query brevi (SQA, Short Query Acceleration).

Chiavi di distribuzione Alter nelle tabelle

Amazon Redshift distribuisce righe di tabelle nel cluster in base allo stile di distribuzione della tabella. Le tabelle con la distribuzione KEY richiedono una colonna come chiave di distribuzione (DISTKEY). La riga di una tabella viene assegnata alla sezione del nodo di un cluster in base al valore della colonna DISTKEY.

Una DISTKEY appropriata posiziona un numero simile di righe su ogni sezione di nodo a cui si fa riferimento di frequente nelle condizioni congiunte. Una condizione congiunta ottimizzata si presenta quando le tabelle vengono unite nelle colonne DISTKEY, accelerando le performance delle query.

Analisi

Il consulente analizza il carico di lavoro del cluster per individuare la chiave di distribuzione più appropriata per le tabelle che possono trarre vantaggi significativi da uno stile di distribuzione KEY.

Raccomandazione

Il consulente fornisce istruzioni ALTER TABLE che alterano il DISTSTYLE e il DISTKEY di una tabella in base alla sua analisi. Per ottenere un vantaggio significativo i termini di performance, devono essere implementate tutte le istruzioni SQL all'interno di un gruppo di raccomandazioni.

La ridistribuzione di una tabella di grandi dimensioni con ALTER TABLE consuma risorse dei cluster e richiede un blocco temporaneo della tabella in momenti diversi. Implementare ogni gruppo di raccomandazioni quando il carico di lavoro dell'altro cluster è leggero. Per maggiori dettagli sull'ottimizzazione delle proprietà di distribuzione delle tabelle, consultare Playbook di progettazione avanzata delle tabelle di engineering di Amazon Redshift: chiavi e stili di distribuzione.

Per ulteriori informazioni su ALTER DISTSYLE e DISTKEY, consultare ALTER TABLE.

> **Note**
>
> Se non viene visualizzato un suggerimento, ciò non significa necessariamente che gli stili di distribuzione correnti siano i più appropriati. Advisor non fornisce consigli quando non ci sono dati sufficienti o quando il vantaggio atteso della ridistribuzione è limitato.
> Le raccomandazioni del consulente si applicano a una tabella particolare e non necessariamente si applicano a una tabella che contiene una colonna con lo stesso nome. Le tabelle che condividono il nome di una colonna possono avere caratteristiche diverse relative a tali colonne, a meno che i dati all'interno della tabella siano gli stessi.
> Se sono presenti raccomandazioni per tabelle provvisorie create o rilasciate da processi ETL, modificare tali processi per utilizzare le chiavi di distribuzione consigliate dal consulente.

Modifica delle chiavi di ordinamento sulle tabelle

Amazon Redshift ordina le righe della tabella in base alla chiave di ordinamento della tabella. L'ordinamento delle righe della tabella si basa sui valori delle colonne chiave di ordinamento.

L'ordinamento di una tabella su una chiave di ordinamento appropriata può accelerare le prestazioni delle query, in particolare quelle con predicati limitati dall'intervallo, richiedendo meno blocchi di tabella da leggere dal disco.

Analisi

Advisor analizza il carico di lavoro del cluster per diversi giorni per identificare una chiave di ordinamento utile per le tabelle.

Raccomandazione

Advisor fornisce due gruppi di istruzioni ALTER TABLE che modificano la chiave di ordinamento di una tabella in base alla sua analisi:

- Istruzioni che modificano una tabella che attualmente non dispone di una chiave di ordinamento per aggiungere una chiave di ordinamento COMPOUND.
- Istruzioni che modificano una chiave di ordinamento da INTERLEAVED a COMPOUND o nessuna chiave di ordinamento.

  L'uso di chiavi dell'ordinamento composto riduce notevolmente l'impegno necessario per la manutenzione. Le tabelle con chiavi di ordinamento composto non richiedono le costose operazioni VACUUM REINDEX necessarie per gli ordinamenti interlacciati. In pratica, le chiavi di ordinamento composto sono più efficaci delle chiavi di ordinamento interlacciato per la maggior parte dei carichi di lavoro Amazon Redshift. Tuttavia, se una tabella è piccola, è più efficiente non avere una chiave di ordinamento per evitare il sovraccarico di archiviazione delle stesse.

Quando si ordina una tabella di grandi dimensioni con ALTER TABLE, le risorse del cluster vengono consumate e i blocchi di tabella sono necessari in momenti diversi. Implementare ogni raccomandazione quando il carico di lavoro di un cluster è moderato. Per maggiori dettagli sull'ottimizzazione delle configurazioni delle chiavi di ordinamento della tabella, consultare Playbook di progettazione avanzata delle tabelle di engineering di Amazon Redshift: chiavi di ordinamento composto e interlacciato.

Per ulteriori informazioni su ALTER SORTKEY, consultare ALTER TABLE.

> **Note**
>
> Se non viene visualizzato un suggerimento per una tabella, ciò non significa necessariamente che la configurazione corrente sia la migliore. Advisor non fornisce consigli quando non ci sono dati sufficienti o quando il vantaggio atteso dell'ordinamento è limitato.
> Le raccomandazioni del consulente si applicano a una tabella particolare e non necessariamente si applicano a una tabella che contiene una colonna con lo stesso nome e lo stesso tipo di dati. Le tabelle che condividono i nomi delle colonne possono avere suggerimenti diversi in base ai dati nelle tabelle e al carico di lavoro.

Modifica delle codifiche di compressione sulle colonne

La compressione è un'operazione a livello di colonna che riduce la dimensione dei dati quando vengono archiviati. La compressione viene utilizzata in Amazon Redshift per risparmiare spazio di archiviazione e migliorare le prestazioni delle query riducendo la quantità di I/O del disco. Consigliamo una codifica di compressione ottimale per ogni colonna in base al tipo di dati e ai modelli di query. Grazie alla compressione ottimale, le query possono essere eseguite in modo più efficiente e il database può occupare uno spazio di archiviazione minimo.

Analisi

Advisor esegue continuamente l'analisi del carico di lavoro e dello schema del database del cluster per identificare la codifica di compressione ottimale per ogni colonna della tabella.

Raccomandazione

Advisor fornisce istruzioni ALTER TABLE che modificano la codifica della compressione di colonne particolari in base alla sua analisi.

La modifica delle codifiche di compressione delle colonne con ALTER TABLE utilizza risorse del cluster e richiede un blocco della tabella in momenti diversi. È consigliabile implementare suggerimenti quando il carico di lavoro del cluster è leggero.

Come riferimento, Esempi di ALTER TABLE mostra diverse istruzioni che modificano la codifica per una colonna.

> **Note**
>
> Advisor non fornisce suggerimenti quando non ci sono dati sufficienti o il vantaggio previsto della modifica della codifica è limitato.

Suggerimenti per i tipi di dati

Amazon Redshift dispone di una libreria di tipi di dati SQL per diversi casi d'uso. Questi includono tipi di numeri interi come `INT` e tipi per memorizzare caratteri, come `VARCHAR`. Inoltre, Redshift memorizza i tipi in modo ottimizzato per fornire accesso rapido e buone prestazioni di query. Inoltre, Redshift fornisce funzioni per tipi specifici che possono essere utilizzate per formattare o eseguire calcoli sui risultati delle query.

Analisi

Advisor esegue continuamente l'analisi del carico di lavoro e dello schema del database del cluster per identificare le colonne che possono trarre vantaggio in modo significativo dalla modifica del tipo di dati.

Raccomandazione

Advisor fornisce un'istruzione `ALTER TABLE` che aggiunge una nuova colonna con il tipo di dati suggerito. Una istruzione `UPDATE` di accompagnamento copia i dati dalla colonna esistente nella nuova colonna. Dopo aver creato la nuova colonna e caricato i dati, modificare le query e gli script di importazione per accedere alla nuova colonna. Quindi sfruttare le funzionalità e le funzioni specializzate per il nuovo tipo di dati, disponibili in Informazioni di riferimento sulle funzioni SQL.

La copia dei dati esistenti nella nuova colonna può richiedere del tempo. Si consiglia di implementare ogni suggerimento di Advisor quando il carico di lavoro del cluster è leggero. Fare riferimento all'elenco dei tipi di dati disponibili all'indirizzo Tipi di dati.

Advisor non fornisce suggerimenti quando non ci sono dati sufficienti o se il vantaggio previsto della modifica della codifica del tipo di dati è limitato.

Tutorial per Amazon Redshift

Seguire la procedura descritta in questi tutorial per conoscere le funzionalità di Amazon Redshift.

- Tutorial: Caricamento dei dati da Amazon S3
- Tutorial: Esecuzione di query su dati nidificati con Amazon Redshift Spectrum
- Tutorial: Configurazione delle code di gestione manuale del carico di lavoro (WLM)
- Tutorial: Utilizzo delle funzioni SQL spaziali con Amazon Redshift
- Tutorial per Amazon Redshift ML

Utilizzo dell'ottimizzazione automatica delle tabelle

L'ottimizzazione automatica delle tabelle è una funzionalità di auto-tuning che ottimizza automaticamente la progettazione delle tabelle applicando le chiavi di ordinamento e distribuzione senza la necessità di un intervento dell'amministratore. Utilizzando l'automazione per ottimizzare la progettazione delle tabelle, è possibile iniziare e ottenere subito le prestazioni più veloci senza investire tempo per ottimizzare e implementare manualmente le ottimizzazioni delle tabelle.

L'ottimizzazione automatica delle tabelle osserva continuamente il modo in cui le query interagiscono con le tabelle. Utilizza metodi avanzati di intelligenza artificiale per scegliere le chiavi di ordinamento e distribuzione per ottimizzare le prestazioni per il carico di lavoro del cluster. Se Amazon Redshift determina che l'applicazione di una chiave migliora le prestazioni del cluster, le tabelle vengono modificate automaticamente entro poche ore dal momento in cui il cluster è stato creato, con un impatto minimo sulle query.

Per sfruttare questa automazione, un amministratore Amazon Redshift crea una nuova tabella o modifica una tabella esistente per consentirgli di utilizzare l'ottimizzazione automatica. Le tabelle esistenti con uno stile di distribuzione o una chiave di ordinamento `AUTO` sono già abilitati per l'automazione. Quando si eseguono query su tali tabelle, Amazon Redshift determina se una chiave di ordinamento o una chiave di distribuzione migliorerà le prestazioni. In tal caso, Amazon Redshift modifica automaticamente la tabella senza richiedere l'intervento dell'amministratore. Se viene eseguito un numero minimo di query, le ottimizzazioni vengono applicate entro poche ore dall'avvio del cluster.

Se Amazon Redshift determina che una chiave di distribuzione migliora le prestazioni delle query, lo stile di distribuzione `AUTO` delle tabelle può cambiare in `KEY`.

Argomenti

Abilitazione dell'ottimizzazione automatica delle tabelle

Per impostazione predefinita, le tabelle create senza definire esplicitamente le chiavi di ordinamento o le chiavi di distribuzione sono impostate su AUTO. Al momento della creazione della tabella, è anche possibile impostare manualmente una chiave di distribuzione o di ordinamento in modo esplicito. Se si imposta la chiave di ordinamento o di distribuzione, la tabella non viene gestita automaticamente.

Per consentire l'ottimizzazione automatica di una tabella esistente, utilizzare le opzioni dell'istruzione ALTER per modificare la tabella in AUTO. È possibile scegliere di definire l'automazione per le chiavi di ordinamento, ma non per le chiavi di distribuzione (e viceversa). Se si esegue un'istruzione ALTER per convertire una tabella in una tabella automatica, le chiavi di ordinamento e gli stili di distribuzione esistenti vengono mantenuti.

```
ALTER TABLE table_name ALTER SORTKEY AUTO;
```

```
ALTER TABLE table_name ALTER DISTSTYLE AUTO;
```

Per ulteriori informazioni, consultare ALTER TABLE.

Inizialmente, una tabella non dispone di una chiave di distribuzione o di una chiave di ordinamento. Lo stile di distribuzione è impostato su EVEN o ALL a seconda delle dimensioni della tabella. Man mano che la tabella cresce di dimensioni, Amazon Redshift applica le chiavi di distribuzione e le chiavi di ordinamento ottimali. Le ottimizzazioni vengono applicate entro poche ore dall'esecuzione di un numero minimo di query. Quando si determinano le ottimizzazioni delle chiavi di ordinamento, Amazon Redshift prova ad ottimizzare i blocchi di dati letti dal disco durante una scansione della tabella. Quando si determinano le ottimizzazioni dello stile di distribuzione, Amazon Redshift prova a ottimizzare il numero di byte trasferiti tra i nodi del cluster.

Rimozione dell'ottimizzazione automatica della tabella da una tabella

È possibile rimuovere la funzione di ottimizzazione automatica da una tabella. La rimozione di una tabella dall'automazione comporta la selezione di una chiave di ordinamento o di uno stile di distribuzione. Per modificare lo stile di distribuzione, specificare uno stile di distribuzione specifico.

```
ALTER TABLE table_name ALTER DISTSTYLE EVEN;
```

```
ALTER TABLE table_name ALTER DISTSTYLE ALL;
```

```
ALTER TABLE table_name ALTER DISTSTYLE KEY DISTKEY c1;
```

Per modificare una chiave di ordinamento, è possibile definire una chiave di ordinamento o sceglierne nessuna.

```
ALTER TABLE table_name ALTER SORTKEY(c1, c2);
```

```
ALTER TABLE table_name ALTER SORTKEY NONE;
```

Monitoraggio delle operazioni di ottimizzazione automatica delle tabelle

La vista di sistema SVV_ALTER_TABLE_RECOMMENDATIONS registra i suggerimenti correnti di Amazon Redshift Advisor per le tabelle. Questa visualizzazione mostra i suggerimenti per tutte le tabelle, indipendentemente dal fatto che siano definite o meno per l'ottimizzazione automatica.

Per verificare se una tabella è definita per l'ottimizzazione automatica, eseguire una query sulla vista di sistema SVV_TABLE_INFO. Le voci sono visualizzate solo per le tabelle visibili nel database della sessione corrente. I suggerimenti vengono inseriti nella vista due volte al giorno a partire già dopo poche ore dal momento della creazione del cluster. Una volta che è disponibile un suggerimento, viene avviata entro un'ora. Dopo che è stato applicato un suggerimento (da Amazon Redshift o dall'utente), non viene più visualizzato nella visualizzazione.

La vista di sistema SVL_AUTO_WORKER_ACTION mostra un log di audit di tutte le operazioni eseguite da Amazon Redshift e lo stato precedente della tabella.

La vista di sistema SVV_TABLE_INFO elenca tutte le tabelle del sistema insieme a una colonna per indicare se la chiave di ordinamento e lo stile di distribuzione della tabella sono impostati su AUTO.

Per ulteriori informazioni sull'utilizzo di queste viste di sistema, consultare Monitoraggio del sistema (solo con provisioning).

Utilizzo della compressione delle colonne

La compressione è un'operazione a livello di colonna che riduce la dimensione dei dati quando vengono archiviati. La compressione preserva lo spazio di storage e riduce le dimensioni dei dati letti dallo storage, diminuendo quindi la quantità di I/O su disco e migliorando le prestazioni delle query.

ENCODE AUTO è l'impostazione di default per le tabelle. Quando la tabella è impostata su ENCODE AUTO, Amazon Redshift gestisce automaticamente la codifica di compressione per tutte le colonne della tabella. Per ulteriori informazioni, consulta CREATE TABLE e ALTER TABLE.

Tuttavia, se si specifica la codifica di compressione per qualsiasi colonna della tabella, la tabella non è più impostata su ENCODE AUTO. Amazon Redshift non gestisce più automaticamente la codifica di compressione per tutte le colonne della tabella.

È possibile applicare un tipo di compressione oppureencoding, alle colonne in una tabella manualmente quando si crea la tabella. In alternativa, è possibile utilizzare il comando COPY per analizzare e applicare automaticamente la compressione. Per ulteriori informazioni, consultare Scelta automatica delle codifiche di compressione con COPY. Per ulteriori informazioni sull'applicazione della compressione automatica, consultare Caricamento di tabelle con compressione automatica.

> **Note**
>
> Consigliamo fortemente l'uso del comando COPY per applicare compressione automatica.

È possibile scegliere di applicare manualmente le codificazioni di compressione nel caso in cui la nuova tabella condivida le stesse caratteristiche di dati di un'altra tabella. In alternativa, è possibile farlo se in fase di verifica si dovesse rilevare che le codifiche di compressione applicate durante la compressione automatica non sono adatte ai propri dati. Se scegli di applicare manualmente le codifiche di compressione, puoi eseguire il comando ANALYZE COMPRESSION su una tabella già popolata e utilizzare i risultati per scegliere codifiche di compressione.

Per applicare manualmente la compressione, specifica le codifiche di compressione per colonne singole come parte della dichiarazione CREATE TABLE. La sintassi è esposta di seguito.

```
CREATE TABLE table_name (column_name
data_type ENCODE encoding-type)[, ...]
```

encoding-type è preso dalla tabella delle parole chiave nella sezione seguente.

Ad esempio, la dichiarazione PRODUCT crea una tabella a due colonne. Quando i dati vengono caricati nella tabella, la colonna PRODUCT_ID non viene compressa; la colonna PRODUCT_NAME viene invece compressa mediante l'uso della codifica del dizionario del byte (BYTEDICT).

```
create table product(
product_id int encode raw,
product_name char(20) encode bytedict);
```

Puoi specificare la codifica di una colonna dopo che viene aggiunta a una tabella mediante l'uso del comando ALTER TABLE.

```
ALTER TABLE table-name ADD [ COLUMN ] column_name column_type ENCODE encoding-type
```

Argomenti

- Codifiche di compressione
- Test delle codifiche di compressione
- Esempio: scelta di codifiche di compressione per la tabella CUSTOMER

Codifiche di compressione

Una codifica di compressione specifica il tipo di compressione applicata a una colonna di valori dei dati nel momento in cui le righe vengono aggiunte a una tabella.

ENCODE AUTO è l'impostazione di default per le tabelle. Quando la tabella è impostata su ENCODE AUTO, Amazon Redshift gestisce automaticamente la codifica di compressione per tutte le colonne della tabella. Per ulteriori informazioni, consulta CREATE TABLE e ALTER TABLE.

Tuttavia, se si specifica la codifica di compressione per qualsiasi colonna della tabella, la tabella non è più impostata su ENCODE AUTO. Amazon Redshift non gestisce più automaticamente la codifica di compressione per tutte le colonne della tabella.

Quando utilizzi CREATE TABLE, ENCODE AUTO è disabilitato quando si specifica la codifica di compressione per qualsiasi colonna della tabella. Amazon Redshift assegna automaticamente una codifica di compressione alle colonne per le quali non si specifica un tipo ENCODE come segue:

- Le colonne definite come chiavi di ordinamento vengono assegnate alla compressione RAW.
- Le colonne definite come tipi di dati BOOLEAN, REAL o DOUBLE PRECISION vengono assegnate alla compressione RAW.

- Le colonne definite come tipo di dati SMALLINT, INTEGER, BIGINT, DECIMAL, CHAR, VARCHAR, DATE, TIMESTAMP o TIMESTAMPTZ sono assegnate alla compressione AZ64.
- Le colonne definite come tipi di dati CHAR o VARCHAR sono assegnate alla compressione LZO.

Puoi modificare la codifica di una tabella dopo averla creata utilizzando ALTER TABLE. Se disabiliti ENCODE AUTO utilizzando ALTER TABLE, Amazon Redshift non gestisce più automaticamente le codifiche di compressione per le colonne. Tutte le colonne manterranno i tipi di codifica di compressione che avevano quando è stato disabilitato ENCODE AUTO finché non si modificano o non si abilita nuovamente ENCODE AUTO.

La seguente tabella identifica le codifiche di compressione supportate, oltre che i tipi di dati che supportano la codifica.

Tipo di codifica	Parole chiave su CREATE TABLE e ALTER TABLE	Tipi di dati
Raw (nessuna compressione)	RAW	Tutti
AZ64	AZ64	SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIMESTAMP, TIMESTAMPTZ
Dizionario byte	BYTEDICT	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMP TZ
Delta	DELTA DELTA32K	SMALLINT, INT, BIGINT, DATE, TIMESTAMP, DECIMAL INT, BIGINT, DATE, TIMESTAMP, DECIMAL
LZO	LZO	SMALLINT, INTEGER, BIGINT, DECIMAL, CHAR, VARCHAR,

Tipo di codifica	Parole chiave su CREATE TABLE e ALTER TABLE	Tipi di dati
		DATE, TIMESTAMP, TIMESTAMP TZ, SUPER
Mostlyn	MOSTLY8	SMALLINT, INT, BIGINT, DECIMAL
	MOSTLY16	INT, BIGINT, DECIMAL
	MOSTLY32	BIGINT, DECIMAL
Run-length	RUNLENGTH	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ
Testo	TEXT255	solo VARCHAR
	TEXT32K	solo VARCHAR
Zstandard	ZSTD	SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ, SUPER

Codifica Raw

La codifica raw è la codifica predefinita per le colonne designate come chiavi di ordinamento, oltre che per le colonne definite come tipi di dati BOOLEAN, REAL o DOUBLE PRECISION. Grazie alla codifica raw, i dati vengono archiviati in una forma non compressa e non elaborata.

Codifica AZ64

AZ64 è un algoritmo di codifica di compressione proprietario progettato da Amazon per ottenere un elevato rapporto di compressione ed elaborazione delle query migliorata. Al suo interno, l'algoritmo AZ64 comprime gruppi più piccoli di valori di dati e utilizza istruzioni SIMD (Single Instruction Multiple

Data) per l'elaborazione parallela. Utilizza AZ64 per ottenere risparmi di storage significativi ed elevate prestazioni per i tipi di dati numerici, data e ora.

Puoi utilizzare AZ64 come algoritmo di codifica di compressione durante la definizione di colonne utilizzando istruzioni CREATE TABLE e ALTER TABLE con i seguenti tipi di dati:

- SMALLINT
- INTEGER
- BIGINT
- DECIMAL
- DATE
- TIMESTAMP
- TIMESTAMPTZ

Codifica del dizionario byte

Nella codifica del dizionario byte, viene creato un dizionario separato da valori univoci per ogni blocco di valori della colonna sul disco. (Un blocco del disco Amazon Redshift occupa 1 MB.) Il dizionario contiene fino a 256 valori da un byte, i quali vengono archiviati come indici dei valori di dati originali. Se in un singolo blocco vengono archiviati più di 256 valori, i valori in eccesso vengono scritti nel blocco in forma non compressa e non elaborata. Lo stesso processo si ripete per ogni blocco del disco.

Questa codifica è molto efficace su colonne di stringhe a bassa cardinalità. Questa codifica è ottimale quando il dominio di dati di una colonna è inferiore a 256 valori univoci.

Per le colonne con il tipo di dati stringa (CHAR e VARCHAR) codificato con BYTEDICT, Amazon Redshift esegue scansioni vettorializzate e valutazioni dei predicati che operano direttamente sui dati compressi. Queste scansioni utilizzano istruzioni singole specifiche dell'hardware e istruzioni con dati multipli (SIMD) per l'elaborazione parallela. Ciò velocizza notevolmente la scansione delle colonne di stringhe. La codifica del dizionario byte è efficiente in termini di spazio, specialmente se una colonna CHAR/VARCHAR contiene lunghe stringhe di caratteri.

Si supponga che una tabella disponga di una colonna COUNTRY con un tipo di dati CHAR(30). Quando i dati vengono caricati, Amazon Redshift crea il dizionario e popola la colonna COUNTRY con il valore indice. Il dizionario contiene i valori univoci indicizzati; inoltre la relativa tabella contiene solo i subscript da un byte dei valori corrispondenti.

> **Note**
>
> Gli spazi iniziali vengono archiviati in colonne di caratteri a lunghezza fissa. Pertanto, in una colonna CHAR(30), ogni valore compresso risparmia 29 byte di storage quando utilizzi la codifica del dizionario byte.

La seguente tabella rappresenta il dizionario per la colonna COUNTRY:

Valori dei dati univoci	Indice del dizionario	Dimensione (lunghezza fissata a 30 byte per valore)
England	0	30
United States of America	1	30
Venezuela	2	30
Sri Lanka	3	30
Argentina	4	30
Japan	5	30
Totale		180

La seguente tabella rappresenta i valori nella colonna COUNTRY:

Valore dati originale	Dimensione originale (lunghezza fissata a 30 byte per valore)	Valore compresso (indice)	Nuova dimensione (byte)
England	30	0	1
England	30	0	1
United States of America	30	1	1

Valore dati originale	Dimensione originale (lunghezza fissata a 30 byte per valore)	Valore compresso (indice)	Nuova dimensione (byte)
United States of America	30	1	1
Venezuela	30	2	1
Sri Lanka	30	3	1
Argentina	30	4	1
Japan	30	5	1
Sri Lanka	30	3	1
Argentina	30	4	1
Totale	300		10

La dimensione totale compressa in questo esempio viene calcolata come di seguito: 6 voci diverse vengono archiviate nel dizionario (6 * 30 = 180), e la tabella contiene 10 valori compressi da un byte, per un totale di 190 byte.

Codifica delta

Le codifiche delta sono molto utili per le colonne date time.

La codifica delta comprime i dati mediante la registrazione della differenza tra i valori che si susseguono nella colonna. Questa differenza viene registrata in un dizionario separato per ogni blocco di valori della colonna sul disco. (Un blocco del disco Amazon Redshift occupa 1 MB.) Ad esempio, si supponga che la colonna contenga 10 interi in una sequenza da 1 a 10. I primi sono memorizzati come un intero da 4 byte (più un flag da 1 byte). I nove successivi vengono memorizzati come byte con il valore 1, indicando che è uno maggiore del valore precedente.

La codifica delta è disponibile in due variazioni:

- DELTA registra le differenze come valori di 1 byte (interi di 8 bit)
- DELTA32K registra le differenze come valori di 2 byte (interi di 16 bit)

Se la maggior parte dei valori nella colonna si potessero comprimere mediante l'uso di un singolo byte, la variazione di 1 byte sarebbe molto efficace. Tuttavia, se i delta sono più grandi, questa codifica, nel peggiore dei casi, è in qualche modo meno efficace rispetto all'archiviazione dei dati non compressi. Una logica simile si applica alla versione 16 bit.

Se la differenza tra i due valori supera l'intervallo di 1 byte (DELTA) o di 2 byte (DELTA32K), il valore completo originale viene archiviato con all'inizio un contrassegno di 1 byte. L'intervallo di 1 byte va da -127 a 127, mentre quello di 2 byte da-32K a 32K.

La tabella seguente mostra come funziona una codifica delta per una colonna numerica:

Valore dati originale	Dimensione originale (byte)	Differenza (delta)	Valore compresso	Dimensione compressa (byte)
1	4		1	1+4 (contrassegno + valore attuale)
5	4	4	4	1
50	4	45	45	1
200	4	150	150	1+4 (contrassegno + valore attuale)
185	4	-15	-15	1
220	4	35	35	1
221	4	1	1	1
Totali	28			15

Codifica LZO

La codifica LZO fornisce un rapporto di compressione molto alto e con buone prestazioni. La codifica LZO funziona particolarmente bene per le colonne CHAR e VARCHAR che archiviano stringhe

di caratteri molto lunghe. Sono particolarmente adatti per testo in formato libero, ad esempio le descrizioni del prodotto, i commenti dell'utente o le stringhe JSON.

Codifica mostly

Le codifiche mostly sono utili quando il tipo di dati di una colonna è maggiore rispetto alla maggior parte dei valori archiviati richiesti. Specificando una codifica mostly per questo tipo di colonna, puoi comprimere la maggior parte dei valori nella colonna in una dimensione di storage standard più piccola. I valori restanti, che non possono essere compressi, vengono archiviati nel loro formato raw. Ad esempio, puoi comprimere una colonna 16 bit, come una colonna INT2, in uno storage 8 bit.

Generalmente, la codifica mostly funziona con i seguenti tipi di dati:

- SMALLINT/INT2 (16 bit)
- INTEGER/INT (32 bit)
- BIGINT/INT8 (64 bit)
- DECIMAL/NUMERIC (64 bit)

Scegli la variazione appropriata della codifica mostly, in funzione della dimensione del tipo di dati per la colonna. Per esempio, applica MOSTLY8 a una colonna definita come colonna intera 16 bit. Non sono consentite applicazioni di MOSTLY16 su una colonna con un tipo di dati 16 bit né applicazioni di MOSTLY32 su una colonna con un tipo di dati 32 bit.

Rispetto all'assenza di compressione, la maggior parte delle codifiche potrebbero essere meno efficaci quando un numero relativamente alto dei valori nella colonna non può essere compresso. Prima di applicare una di queste codificazioni a una colonna, eseguire un controllo. La maggior parte dei valori che stanno per essere caricati (e che saranno caricati in seguito) dovrebbe adattarsi agli intervalli riportati nella seguente tabella.

Encoding	Dimensione storage compressa	L'intervallo dei valori che può essere compresso (i valori fuori dall'intervallo vengono archiviati in formato raw)
MOSTLY8	1 byte (8 bit)	Da -128 a 127
MOSTLY16	2 byte (16 bit)	Da -32768 a 32767
MOSTLY32	4 byte (32 bit)	Da -2147483648 a +2147483647

> **Note**
>
> Per i valori decimali, ignora la posizione del punto per determinare se il valore è appropriato all'intervallo. Ad esempio, 1.234.56 viene considerato come 123.456 e può essere compresso in una colonna MOSTLY32.

Ad esempio, la colonna VENUEID sulla tabella VENUE viene definita come una colonna intera in formato raw, ciò significa che il suo valore consuma 4 byte dello storage. Ad ogni modo, l'intervallo attuale dei valori nella colonna va da **`0`** a **`309`**. Quindi ricreare e ricaricare questa tabella con la codifica MOSTLY16 per VENUEID ridurrebbe a 2 byte l'archiviazione di ogni valore in quella colonna.

Se la maggior parte dei valori VENUEID a cui si fa riferimento in un'altra tabella fossero nell'intervallo da 0 a 127, avrebbe senso codificare la colonna della chiave esterna come MOSTLY8. Prima di scegliere, sarà necessario eseguire varie query sui dati della tabella di riferimento, per scoprire se la maggior parte dei valori rientra nell'intervallo a 8 bit, 16 bit o 32 bit.

La tabella seguente mostra le dimensioni compresse per i valori numerici specifici, al momento dell'utilizzo delle codifiche MOSTLY8, MOSTLY16 e MOSTLY32:

Valore originale	Dimensione originale INT o BIGINT (byte)	Dimensione compressa MOSTLY8 (byte)	Dimensione compressa MOSTLY16 (byte)	Dimensione compressa MOSTLY32 (byte)
1	4	1	2	4
10	4	1	2	4
100	4	1	2	4
1000	4	La stessa dimensione dei dati raw	2	4
10000	4		2	4
20000	4		2	4
40000	8		La stessa dimensione dei dati raw	4

Valore originale	Dimensione originale INT o BIGINT (byte)	Dimensione compressa MOSTLY8 (byte)	Dimension e compressa MOSTLY16 (byte)	Dimension e compressa MOSTLY32 (byte)
100000	8			4
2000000000	8			4

Codifica della lunghezza di esecuzione

La codifica della lunghezza di esecuzione sostituisce un valore che viene ripetuto consecutivamente con un token, il quale è composto dal valore e dal conteggio del numero delle ricorrenze consecutive (la lunghezza dell'esecuzione). Viene creato un dizionario separato da valori univoci per ogni blocco di valori della colonna sul disco. (Un blocco del disco Amazon Redshift occupa 1 MB.) Questa codifica è la più appropriata per una tabella in cui i valori dei dati vengono spesso ripetuti consecutivamente, per esempio quando la tabella è ordinata in base a quei valori.

Ad esempio, si supponga che una colonna in una tabella di grandi dimensioni disponga di un dominio abbastanza piccolo, ad esempio una colonna COLOR con meno di 10 valori possibili. È probabile che questi valori cadano in sequenze lunghe in tutta la tabella, anche se i dati non sono ordinati.

Consigliamo di non applicare la codifica della lunghezza di esecuzione su nessuna colonna indicata come chiave di ordinamento. Le scansione a intervallo limitato hanno una migliore prestazione quando i blocchi contengono numeri di righe simili. Se le colonne della chiave di ordinamento vengono compresse più delle altre colonne nella stessa query, le prestazioni delle scansioni a intervallo limitato potrebbero essere scarse.

La tabella seguente usa l'esempio della colonna COLOR per mostrare come funziona la codifica di lunghezza dell'esecuzione:

Valore dati originale	Dimensione originale (byte)	Valore compresso (token)	Dimensione compressa (byte)
Blue	4	{2,Blue}	5
Blue	4		0

Valore dati originale	Dimensione originale (byte)	Valore compresso (token)	Dimensione compressa (byte)
Green	5	{3,Green}	6
Green	5		0
Green	5		0
Blue	4	{1,Blue}	5
Yellow	6	{4,Yellow}	7
Yellow	6		0
Yellow	6		0
Yellow	6		0
Totale	51		23

Codifiche Text255 e Text32k

Le codifiche Text255 e Text32k sono utili per la compressione delle colonne VARCHAR in cui ricorrono spesso le stesse parole. Viene creato un dizionario separato da parole univoche per ogni blocco di valori della colonna sul disco. (Un blocco del disco Amazon Redshift occupa 1 MB.) Il dizionario contiene le prime 245 parole univoche nella colonna. Queste parole vengono sostituite sul disco da un valore dell'indice 1 byte, che rappresenta uno dei 245 valori; inoltre tutte le parole che non sono nel dizionario vengono archiviate senza essere compresse. Lo stesso processo si ripete per ogni blocco del disco da 1 MB. Se le parole indicizzate si verificano frequentemente nella colonna, questa produrrà un alto rapporto di compressione.

Per quando riguarda la codifica text32k, il principio è lo stesso ma il dizionario di ogni blocco non acquisisce uno specifico numero di parole. Al contrario, il dizionario indicizza ogni parola univoca che rileva finché le voci combinate raggiungono una lunghezza di 32k, meno qualche spesa di gestione. I valori dell'indice vengono archiviati in due byte.

Considera, ad esempio, la colonna VENUENAME nella tabella VENUE. Parole come **`Arena`**, **`Center`** e **`Theatre`** sono ricorrenti in questa colonna e, probabilmente, sarebbero tra le prime 245 parole

trovate in ogni blocco se fosse applicata la compressione text255. In tal caso, questa colonna beneficia della compressione. Questo perché ogni volta che appariranno queste parole, occuperanno solo 1 byte dello storage (anziché 5, 6 o 7 rispettivamente).

Codifica Zstandard

La codifica Zstandard (ZSTD) fornisce un elevato rapporto di compressione con buone prestazioni tra i diversi set di dati. La codifica ZSTD funziona particolarmente bene con le colonne CHAR e VARCHAR che archiviano un'ampia gamma di stringhe lunghe e corte, ad esempio le descrizioni del prodotto, i commenti dell'utente, i log e le stringhe JSON. Mentre alcuni algoritmi, come la codifica Delta o la codifica Mostly, potenzialmente possono utilizzare più spazio di storage rispetto all'assenza di compressione, è molto improbabile che la codifica ZSTD incrementi l'utilizzo del disco.

ZSTD supporta i tipi di dati SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP e TIMESTAMPTZ.

Test delle codifiche di compressione

Nel caso in cui decidessi di specificare manualmente le codifiche della colonna, probabilmente vorresti eseguire i test delle diverse codifiche dei tuoi dati.

> **Note**
>
> Se possibile, ti consigliamo l'utilizzo del comando COPY per caricare i dati e per permettergli di scegliere le codifiche ottimali a seconda dei tuoi dati. In alternativa, è possibile utilizzare il comando ANALYZE COMPRESSION per visualizzare le codifiche consigliate per i dati esistenti. Per ulteriori informazioni sull'applicazione della compressione automatica, consultare Caricamento di tabelle con compressione automatica.

Per eseguire un test della compressione di dati significativo, sarà necessario un gran numero di righe. In questo esempio, verrà creata una tabella e saranno inserite delle righe mediante l'uso di una dichiarazione che seleziona dati da due tabelle, VENUE e LISTING. Lasciamo fuori la clausola WHERE che normalmente unirebbe le due tabelle. Il risultato è che ogni riga nella tabella VENUE viene unita a tutte le righe nella tabella LISTING, per un totale di più di 32 milioni di righe. Questa operazione, conosciuta come join cartesiano, solitamente non è consigliata. Tuttavia, a questo scopo, è un metodo conveniente per la creazione di molte righe. Se disponi di una tabella esistente con dei dati su cui desideri eseguire dei test, puoi ignorare questa fase.

Una volta ottenuta una tabella con dati di esempio, viene creata una tabella con sette colonne. Ognuna di esse ha una diversa codifica di compressione: raw, bytedict, lzo, lunghezza di esecuzione, text255, text32k e zstd. Ogni colonna viene completata con esattamente gli stessi dati eseguendo un comando INSERT che seleziona i dati dalla prima tabella.

Per testare le codifiche di compressione, procedere come indicato di seguito:

1. (Facoltativo) Per prima cosa, utilizzare un join cartesiano al fine di creare una tabella con un gran numero di righe. Salta questa fase se desideri eseguire il test di una tabella esistente.

```
create table cartesian_venue(
venueid smallint not null distkey sortkey,
venuename varchar(100),
venuecity varchar(30),
venuestate char(2),
venueseats integer);

insert into cartesian_venue
select venueid, venuename, venuecity, venuestate, venueseats
from venue, listing;
```

2. Successivamente, creare una tabella con le codifiche che si desidera confrontare.

```
create table encodingvenue (
venueraw varchar(100) encode raw,
venuebytedict varchar(100) encode bytedict,
venuelzo varchar(100) encode lzo,
venuerunlength varchar(100) encode runlength,
venuetext255 varchar(100) encode text255,
venuetext32k varchar(100) encode text32k,
venuezstd varchar(100) encode zstd);
```

3. Inserire gli stessi dati in tutte le colonne utilizzando la dichiarazione INSERT con una clausola SELECT.

```
insert into encodingvenue
select venuename as venueraw, venuename as venuebytedict, venuename as venuelzo,
 venuename as venuerunlength, venuename as  venuetext32k, venuename as  venuetext255,
 venuename as venuezstd
from cartesian_venue;
```

4. Verificare il numero di righe nella nuova tabella.

```
select count(*) from encodingvenue

  count
----------
 38884394
(1 row)
```

5. Eseguire una query della tabella del sistema STV_BLOCKLIST per confrontare il numero di blocchi del disco da 1 MB utilizzati da ogni colonna.

   La funzione di aggregazione MAX restituisce il numero massimo di blocchi per ogni colonna. La tabella STV_BLOCKLIST include i dettagli delle tre colonne generate dal sistema. Questo esempio utilizza `col < 6` nella clausola WHERE per escludere le colonne generate dal sistema.

```
select col, max(blocknum)
from stv_blocklist b, stv_tbl_perm p
where (b.tbl=p.id) and name ='encodingvenue'
and col < 7
group by name, col
order by col;
```

   La query restituisce i seguenti risultati. Le colonne sono numerate a partire da zero. A seconda della configurazione del tuo cluster, i risultati potrebbero avere numeri diversi, ma le dimensioni relative dovrebbero essere simili. È possibile osservare che la codifica BYTEDICT sulla seconda colonna ha prodotto i migliori risultati per questo set di dati. Questo approccio ha un rapporto di compressione migliore di 20:1. Anche le codifiche LZO e ZSTD hanno prodotto risultati eccellenti. Set di dati diversi produrranno naturalmente risultati diversi. Se una colonna contiene stringhe di testo più lunghe, la codifica LZO spesso produce i migliori risultati di compressione.

```
 col | max
-----+-----
   0 | 203
   1 |  10
   2 |  22
   3 | 204
   4 |  56
   5 |  72
   6 |  20
(7 rows)
```

Se disponi di dati in una tabella esistente, puoi utilizzare il comando ANALYZE COMPRESSION per visualizzare le codifiche consigliate per la tabella. Ad esempio, l'esempio seguente mostra la codifica consigliata per una copia della tabella VENUE, ovvero CARTESIAN_VENUE, che contiene 38 milioni di righe. Nota che ANALYZE COMPRESSION consiglia la codifica LZO per la colonna VENUENAME. ANALYZE COMPRESSION sceglie la compressione ottimale secondo diversi fattori, tra cui una percentuale di riduzione. In questo caso specifico, BYTEDICT fornisce una migliore compressione, ma anche LZO produce una compressione maggiore del 90 percento.

```
analyze compression cartesian_venue;

Table          | Column     | Encoding | Est_reduction_pct
---------------+------------+----------+------------------
reallybigvenue | venueid    | lzo      | 97.54
reallybigvenue | venuename  | lzo      | 91.71
reallybigvenue | venuecity  | lzo      | 96.01
reallybigvenue | venuestate | lzo      | 97.68
reallybigvenue | venueseats | lzo      | 98.21
```

Esempio: scelta di codifiche di compressione per la tabella CUSTOMER

La seguente dichiarazione crea una tabella CUSTOMER che dispone di colonne con diversi tipi di dati. Questa dichiarazione CREATE TABLE mostra una delle possibili combinazioni delle codifiche di compressione per queste colonne.

```
create table customer(
custkey int encode delta,
custname varchar(30) encode raw,
gender varchar(7) encode text255,
address varchar(200) encode text255,
city varchar(30) encode text255,
state char(2) encode raw,
zipcode char(5) encode bytedict,
start_date date encode delta32k);
```

La tabella seguente mostra le codifiche della colonna che sono state scelte per la tabella CUSTOMER e fornisce una spiegazione della scelta:

Colonna	Tipo di dati	Encoding	Spiegazione
CUSTKEY	int	delta	CUSTKEY è composta da valori interi consecutivi e univoci. Dato che le differenze saranno di 1 byte, la codifica DELTA è una buona scelta.
CUSTNAME	varchar(30)	raw	CUSTNAME dispone di un grande dominio con pochi valori ripetuti. Qualunque codifica di compressi one probabilmente non sarebbe efficace.
GENDER	varchar(7)	text255	GENDER è un dominio molto piccolo con molti valori ripetuti. Text255 funziona bene con le colonne VARCHAR in cui ricorrono le stesse parole.
ADDRESS	varchar(200)	text255	ADDRESS è un dominio grande, ma contiene molte parole ripetute come Street, Avenue, North, South e così via. Le codifiche Text255 e text32k sono utili per la compressione delle

Colonna	Tipo di dati	Encoding	Spiegazione
			colonne VARCHAR in cui ricorrono le stesse parole. La lunghezza della colonna è corta, pertanto text255 è una buona scelta.
CITY	varchar(30)	text255	CITY è un dominio grande, con alcuni valori ripetuti. Alcuni nomi di città sono utilizzati di più rispetto ad altri. Text255 è una buona scelta per lo stesso motivo di ADDRESS.
STATE	char(2)	raw	Negli Stati Uniti, STATE è un dominio preciso di 50 valori da due caratteri. La codifica bytedict restituirebbe delle compressioni, ma siccome la dimension e della colonna è di soli due caratteri , la compressione potrebbe non essere conveniente per i costi di gestione relativi alla decompressione dei dati.

Colonna	Tipo di dati	Encoding	Spiegazione
ZIPCODE	char(5)	bytedict	ZIPCODE è un dominio noto di poco più di 50.000 valori univoci. Alcuni codici zip ricorrono più di altri. La codifica bytedict è molto efficace quando una colonna contiene un numero limitato di valori univoci.
START_DATE	data	delta32k	Le codifiche delta sono molto utili per le colonne date time, in particolare quando le righe vengono caricate nell'ordine di data.

Utilizzo degli stili di distribuzione dati

Quando si caricano i dati in una tabella, Amazon Redshift distribuisce le righe della tabella su tutti i nodi di calcolo a seconda dello stile di distribuzione della tabella. Quando si esegue una query, l'ottimizzatore di query ridistribuisce le righe sui nodi di calcolo per eseguire qualsiasi operazione di join e aggregazione, in base alle necessità. La selezione di uno stile di distribuzione di tabella ha l'obiettivo di ridurre al minimo l'impatto della fase di ridistribuzione posizionando i dati dove necessario prima dell'esecuzione della query.

> **Note**
>
> In questa sezione verranno presentati i principi della distribuzione dei dati in un database Amazon Redshift. È consigliabile creare le tabelle con `DISTSTYLE AUTO`. In tal caso, Amazon Redshift utilizza l'ottimizzazione automatica della tabella per scegliere lo stile

di distribuzione dei dati. Per ulteriori informazioni, consultare Utilizzo dell'ottimizzazione automatica delle tabelle. Il resto di questa sezione fornisce dettagli sugli stili di distribuzione.

Argomenti

Concetti della distribuzione dei dati

Di seguito sono riportati alcuni concetti di distribuzione dei dati per Amazon Redshift.

Nodi e sezioni

Un cluster Amazon Redshift è un set di nodi. Ogni nodo nel cluster ha il proprio sistema operativo, una memoria dedicata e uno storage del disco dedicato. Uno dei nodi è il nodo principale, il quale gestisce la distribuzione dei dati ed esegue query delle attività di elaborazione ai nodi di calcolo. I nodi di calcolo forniscono risorse per eseguire tali attività.

Lo storage del disco di un nodo di calcolo è diviso in un numero di sezioni. Il numero di sezioni per nodo dipende dalla dimensione dei nodi del cluster. Tutti i nodi partecipano all'esecuzione di query parallele, lavorando sui dati di calcolo distribuiti più uniformemente possibile all'interno delle sezioni. Per ulteriori informazioni sul numero di sezioni per ogni dimensione di nodo, consulta Informazioni su cluster e nodi nella Guida alla gestione di Amazon Redshift.

Ridistribuzione dei dati

Quando si caricano i dati in una tabella, Amazon Redshift distribuisce le righe della tabella su ognuna delle sezioni dei nodi a seconda dello stile di distribuzione della tabella. Come parte di un piano di query, l'ottimizzatore determina dove è necessario posizionare i blocchi dei dati per la migliore esecuzione della query. I dati quindi vengono spostati fisicamente, o ridistribuiti, durante l'esecuzione

della query. La redistribuzione potrebbe comportare l'invio di specifiche righe ai nodi per il join o la diffusione di un'intera tabella a tutti i nodi.

La ridistribuzione dei dati può contare su una porzione sostanziale dei costi del piano di query; inoltre, il traffico di rete generato dalla ridistribuzione dei dati, può influire sulle altre operazioni del database e può rallentare le prestazioni generali del sistema. Qualora prevedessi dove converrebbe posizionare inizialmente i dati, potrai minimizzare l'impatto della ridistribuzione dei dati.

Obiettivi della distribuzione dei dati

Quando si caricano i dati in una tabella, Amazon Redshift distribuisce le righe della tabella su tutti i nodi di calcolo e alle sezioni, a seconda dello stile di distribuzione scelto al momento della creazione della tabella. La distribuzione dei dati ha due obiettivi principali:

- Distribuire il carico di lavoro uniformemente su tutti i nodi del cluster. La distribuzione di dati non uniforme, o la differenza di distribuzione dei dati, forza alcuni nodi a lavorare maggiormente rispetto ad altri e compromette le prestazioni della query.
- Minimizzare lo spostamento dei dati durante l'esecuzione di una query. Se le righe che partecipano a join o alle aggregazioni si trovano già sui nodi con le loro righe di join in altre tabelle, l'ottimizzatore non dovrà ridistribuire molti dati durante l'esecuzione della query.

La strategia di distribuzione che hai scelto per il tuo database avrà importanti ripercussioni sulle prestazioni della query, sui requisiti dello storage, sul caricamento dei dati e sulla manutenzione. Scegliendo lo stile di distribuzione migliore per ogni tabella, puoi bilanciare la distribuzione dei dati e migliorare in modo significativo le prestazioni generali del sistema.

Stili di distribuzione

Quando create una tabella, potete designare uno dei seguenti stili di distribuzione: AUTO, EVEN, KEY o ALL.

Se non si specifica uno stile di distribuzione, Amazon Redshift utilizza la distribuzione AUTO.

Distribuzione AUTO

Con la distribuzione AUTO, Amazon Redshift assegna uno stile di distribuzione ottimale basato sulla dimensione dei dati della tabella. Ad esempio, se viene specificato lo stile di distribuzione AUTO, Amazon Redshift assegna inizialmente la distribuzione ALL a una tabella di piccole dimensioni. Quando le dimensioni della tabella aumentano, Amazon Redshift potrebbe modificare lo stile di

distribuzione in KEY, scegliendo la chiave primaria (o una colonna della chiave primaria composita) come chiave di distribuzione. Se le dimensioni della tabella aumentano e nessuna delle colonne è adatta per essere utilizzata come chiave di distribuzione, Amazon Redshift modifica lo stile di distribuzione in EVEN. La modifica dello stile di distribuzione avviene in background con un impatto minimo sulle query degli utenti.

Per visualizzare le operazioni eseguite automaticamente da Amazon Redshift per modificare una chiave di distribuzione della tabella, consultare SVL_AUTO_WORKER_ACTION. Per visualizzare i suggerimenti correnti relativi alla modifica di una chiave di distribuzione della tabella, consultare SVV_ALTER_TABLE_RECOMMENDATIONS.

Per visualizzare lo stile di distribuzione applicato a una tabella, eseguire una query sulla visualizzazione del catalogo di sistema PG_CLASS_INFO. Per ulteriori informazioni, consultare Visualizzazione degli stili di distribuzione. Se non si specifica uno stile di distribuzione con l'istruzione CREATE TABLE, Amazon Redshift applica la distribuzione AUTO.

Distribuzione EVEN

Il nodo principale distribuisce le righe tra le sezioni con un metodo round robin, indipendentemente dai valori in una determinata colonna. La distribuzione EVEN è appropriata quando una tabella non partecipa ai join. È appropriata anche quando non c'è una scelta chiara tra la distribuzione KEY e la distribuzione ALL.

Distribuzione KEY

Le righe sono distribuite in funzione dei valori in una colonna. Il nodo principale posiziona valori corrispondenti sulla stessa sezione di nodo. Se si distribuisce una coppia di tabelle sulle chiavi di join, il nodo principale colloca le righe sulle sezioni in funzione dei valori nelle colonne di join affinché i valori corrispondenti delle colonne comuni siano archiviati fisicamente insieme. In questo modo, i valori corrispondenti delle colonne comuni vengono fisicamente memorizzati insieme.

Distribuzione ALL

Un copia dell'intera tabella viene distribuita a ogni nodo. Dove la distribuzione EVEN o KEY posiziona solo una parte delle righe di una tabella su ogni nodo, la distribuzione ALL assicura la collocazione di ogni riga per ogni join a cui la tabella partecipa.

La distribuzione ALL moltiplica lo storage richiesto dal numero di nodi nel cluster, quindi impiega più tempo per caricare, aggiornare o inserire i dati in più tabelle. La distribuzione ALL è idonea solo per spostamenti relativamente lenti delle tabelle, ovvero per le tabelle che non vengono aggiornate

intensamente o frequentemente. Poiché il costo della ridistribuzione di piccole tabelle durante una query è basso, non vi è un vantaggio significativo per definire le tabelle di dimensioni ridotte come DISTSTYLE ALL.

> **Note**
>
> Dopo aver specificato uno stile di distribuzione per una colonna, Amazon Redshift gestisce la distribuzione dei dati a livello di cluster. Amazon Redshift non richiede né supporta il concetto di partizionamento dei dati all'interno degli oggetti del database. Non è necessario creare spazi di tabelle o definire schemi di partizionamento per le tabelle.

In alcuni scenari, puoi modificare lo stile di distribuzione di una tabella dopo che è stata creata. Per ulteriori informazioni, consultare ALTER TABLE. Per gli scenari in cui non puoi modificare lo stile di distribuzione di una tabella dopo che è stata creata, puoi ricreare la tabella e popolarla con una copia completa. Per ulteriori informazioni, consultare Esecuzione di una copia completa

Visualizzazione degli stili di distribuzione

Per visualizzare lo stile di distribuzione di una tabella, eseguire una query sulla visualizzazione PG_CLASS_INFO o SVV_TABLE_INFO.

La colonna RELEFFECTIVEDISTSTYLE in PG_CLASS_INFO indica lo stile di distribuzione attuale per la tabella. Se la tabella utilizza la distribuzione automatica, RELEFFECTIVEDISTSTYLE è 10, 11 o 12 che indica se lo stile di distribuzione effettivo è AUTO (ALL), AUTO (EVEN) o AUTO (KEY). Se la tabella utilizza la distribuzione automatica, lo stile di distribuzione potrebbe inizialmente mostrare AUTO (ALL), quindi passare ad AUTO (EVEN) o AUTO (KEY) quando le dimensioni della tabella aumentano.

La seguente tabella fornisce lo stile di distribuzione per ogni valore su RELEFFECTIVEDISTSTYLE:

RELEFFECTIVEDISTSTYLE	Stile di distribuzione attuale
0	EVEN
1	KEY
8	ALL

RELEFFECTIVEDISTSTYLE	Stile di distribuzione attuale
10	AUTO (ALL)
11	AUTO (EVEN)
12	AUTO (KEY)

La colonna DISTSTYLE in SVV_TABLE_INFO indica lo stile di distribuzione attuale per la tabella. Se la tabella utilizza la distribuzione automatica, DISTSTYLE è AUTO (ALL), AUTO (EVEN) o AUTO (KEY).

L'esempio seguente crea quattro tabelle mediante l'uso di tre stili di distribuzione e distribuzione automatica, quindi esegue una query SVV_TABLE_INFO per visualizzare gli stili di distribuzione.

```
create table public.dist_key (col1 int)
diststyle key distkey (col1);

insert into public.dist_key values (1);

create table public.dist_even (col1 int)
diststyle even;

insert into public.dist_even values (1);

create table public.dist_all (col1 int)
diststyle all;

insert into public.dist_all values (1);

create table public.dist_auto (col1 int);

insert into public.dist_auto values (1);

select "schema", "table", diststyle from SVV_TABLE_INFO
where "table" like 'dist%';

        schema   |    table        | diststyle
     ------------+-----------------+------------
      public     | dist_key        | KEY(col1)
```

```
public     | dist_even       | EVEN
public     | dist_all        | ALL
public     | dist_auto       | AUTO(ALL)
```

Valutazione dei modelli di query

La scelta di stili di distribuzione è solo uno degli aspetti della progettazione del database. Si dovrebbero prendere in considerazione gli stili di distribuzione solo all'interno del contesto dell'intero sistema, bilanciando la distribuzione con altri importanti fattori, come la dimensione dei cluster, i metodi di codifica della compressione, le chiavi di ordinamento e i limiti della tabella.

Esegui dei test sul tuo sistema con dei dati che siano più reali possibili.

Per scegliere lo stile di distribuzione migliore, sarà necessario capire i modelli di query per l'applicazione Amazon Redshift. Identifica le query più costose nel tuo sistema e basa il progetto iniziale del database sulle richieste di queste query. I fattori che determinano il costo totale di una query sono i tempi di esecuzione della query e i relativi consumi di risorse di calcolo. Altri fattori che determinano il costo della query sono i tempi di esecuzione e quanto le altre query e le operazioni del database vengono rivoluzionate.

Identificare le tabelle utilizzate dalle query più costose e valutare il loro ruolo nel runtime delle query. Considera il modo in cui le tabelle vengono combinate e aggregate.

Utilizza le linee guida di questa sezione per scegliere uno stile di distribuzione per ogni tabella. Dopo averlo fatto, creare le tabelle e caricarle con dei dati che siano più reali possibile. Quindi testare le tabelle per i tipi di query che si prevede di utilizzare. Puoi valutare i piani di illustrazione di query per identificare le opportunità di ottimizzazione. Confrontare i tempi di caricamento, lo spazio di archiviazione e i tempi di esecuzione della query per bilanciare i requisiti generali del sistema.

Indicazione degli stili di distribuzione

In questa sezione, le considerazioni e le raccomandazioni per l'indicazione degli stili di distribuzione utilizzano uno schema a stella come esempio. Il progetto del database potrebbe essere basato su uno star schema, su alcune sue varianti o su uno schema completamente diverso. Amazon Redshift è progettato per funzionare in modo efficace con qualsiasi schema di progettazione scelto. I principi di questa sezione possono essere applicati a qualsiasi schema di progetto.

1. Specificare la chiave primaria e le chiavi esterne per tutte le tabelle.

Amazon Redshift non applica limitazioni di chiavi primarie o esterne, ma l'ottimizzatore di query le utilizza quando genera i piani di query. Se imposti le chiavi primarie e quelle esterne, la tua applicazione dovrà mantenere la validità delle chiavi.

2. Distribuire le tabelle dei fatti e le sue tabelle di dimensioni più grandi sulle loro colonne comuni.

   Scegli quella di dimensioni più grandi a seconda della dimensione del set di dati coinvolto nel join più comune, non solo per la dimensione della tabella. Se una tabella di solito viene filtrata mediante l'utilizzo della clausola WHERE, solo una parte delle sue righe parteciperà alla combinazione. Tabelle del genere hanno un impatto minore sulla ridistribuzione rispetto a tabelle più piccole che forniscono più dati. Indica sia la dimensione della chiave primaria della tabella che la chiave esterna corrispondente della tabella dei fatti come DISTKEY. Se più tabelle utilizzano la stessa chiave di distribuzione, verranno posizionate con la tabella dei fatti. La tabella dei fatti può avere una sola chiave di distribuzione. Le tabelle che eseguono l'operazione join su un'altra chiave non sono collocate con la tabella dei fatti.

3. Indicare le chiavi di distribuzione per le altre tabelle dimensionali.

   Distribuisci le tabelle sulle loro chiavi primarie o esterne, a seconda di come combinano più frequentemente con le altre tabelle.

4. Valutare se sia il caso di modificare alcune tabelle dimensionali per utilizzare la dimensione ALL.

   Se una tabella di dimensioni non può essere collocata con la tabella dei fatti o altre importanti tabelle di join, è spesso possibile migliorare le prestazioni delle query in modo significativo distribuendo l'intera tabella su tutti i nodi. L'utilizzo della distribuzione ALL moltiplica i requisiti di spazio di storage e aumenta i tempi di caricamento oltre che le operazioni di manutenzione; è quindi necessario valutare tutti i fattori prima di scegliere la distribuzione ALL. La seguente sezione illustra come identificare i candidati per la distribuzione ALL, mediante la valutazione del piano EXPLAIN.

5. Utilizzare la distribuzione AUTO per le tabelle restanti.

   Se una tabella è ampiamente denormalizzata e non partecipa ai join, oppure se non hai ben chiara la scelta di un altro stile di distribuzione, utilizza la distribuzione AUTO.

Per permettere ad Amazon Redshift di scegliere lo stile di distribuzione appropriato, non specificare esplicitamente una distribuzione automatica.

Valutazione del piano di query

Puoi utilizzare i piani di query per identificare i candidati per l'ottimizzazione dello stile di distribuzione.

Dopo avere fatto la scelta iniziale, crea le tue tabelle, caricale con i dati e verificale. Utilizza un set di dati per il test che sia più reale possibile. Misura i tempi di caricamento per utilizzarli come riferimento per il confronto.

Valutare le query che sono rappresentative di quelle più costose che verranno eseguite; nello specifico, le query che utilizzano combinazioni e aggregazioni. Confrontare i runtime per le varie opzioni di progettazione. Quando si confrontano i runtime di una query, non tenere in conto il primo runtime della query perché include il tempo di compilazione.

DS_DIST_NONE

Non è necessaria alcuna ridistribuzione, perché le sezioni corrispondenti vengono posizionate sui nodi di calcolo. Solitamente si avrà una sola fase DS_DIST_NONE, la combinazione tra la tabella dei fatti e una tabella dimensionale.

DS_DIST_ALL_NONE

Non è necessaria alcuna ridistribuzione, perché la tabella di combinazione interna utilizza DISTSTYLE ALL. L'intera tabella si trova su ogni nodo.

DS_DIST_INNER

La tabella interna viene ridistribuita.

DS_DIST_OUTER

La tabella esterna viene ridistribuita.

DS_BCAST_INNER

Una copia di tutta la tabella interna viene trasmessa a tutti i nodi di calcolo.

DS_DIST_ALL_INNER

Tutta la tabella interna viene ridistribuita in una singola sezione, perché la tabella esterna utilizza DISTSTYLE ALL.

DS_DIST_BOTH

Entrambe le tabelle vengono ridistribuite.

DS_DIST_NONE e DS_DIST_ALL_NONE vanno bene. Indicano che non è stata necessaria alcuna ridistribuzione per quella fase, perché tutte le combinazioni erano posizionate.

DS_DIST_INNER significa che la fase probabilmente avrà un costo relativamente alto, perché la tabella interna è stata ridistribuita sui nodi. DS_DIST_INNER indica che la tabella esterna è già stata propriamente distribuita sulla chiave di combinazione. Imposta la chiave di distribuzione della tabella interna sulla chiave di combinazione per convertirla in DS_DIST_NONE. In alcuni casi, la distribuzione della tabella interna sulla chiave di combinazione non è possibile, perché la tabella esterna non è distribuita sulla chiave di combinazione. In questo caso, valutare se utilizzare la distribuzione ALL per la tabella interna. Se la tabella non viene aggiornata frequentemente o ampiamente, oltre a essere troppo grande per portare alti costi di ridistribuzione, modificare lo stile di distribuzione su ALL e riprovare. Le clausole di distribuzione ALL incrementano i tempi di caricamento, quindi se esegui nuovi test, includi il tempo di caricamento nei fattori di valutazione.

DS_DIST_ALL_INNER non va bene. Ciò significa che tutta la tabella interna viene ridistribuita in una singola sezione, perché la tabella esterna utilizza DISTSTYLE ALL, in modo che una copia di tutta la tabella esterna venga posizionata su ogni nodo. Questo comporta un'esecuzione seriale inefficiente della combinazione su un singolo nodo, anziché trarre vantaggio dal runtime parallelo che utilizza tutti i nodi. DISTSTYLE ALL dovrebbe essere utilizzata solo per la tabella di combinazione interna. Per la tabella esterna invece, specifica una chiave di distribuzione o utilizza la distribuzione EVEN.

DS_BCAST_INNER e DS_DIST_BOTH non vanno bene. Solitamente queste ridistribuzioni si verificano perché le tabelle non vengono combinate sulle loro chiavi di ridistribuzione. Se la tabella dei fatti non dispone ancora di una chiave di distribuzione, specifica la colonna di combinazione come la chiave di distribuzione per entrambe le tabelle. Se la tabella dei fatti ha già una chiave di distribuzione su un'altra colonna, valutare se la modifica della chiave di distribuzione per collocare questo join migliora le prestazioni complessive. Se la modifica della chiave di distribuzione della tabella esterna non dovesse essere una scelta ottimale, è possibile ottenere la collocazione specificando DISTSTYLE ALL per tutte le tabelle interne.

L'esempio seguente mostra una parte del piano di query con le etichette DS_BCAST_INNER e DS_DIST_NONE.

```
->  XN Hash Join DS_BCAST_INNER  (cost=112.50..3272334142.59 rows=170771 width=84)
        Hash Cond: ("outer".venueid = "inner".venueid)
        ->  XN Hash Join DS_BCAST_INNER  (cost=109.98..3167290276.71 rows=172456
 width=47)
              Hash Cond: ("outer".eventid = "inner".eventid)
              ->  XN Merge Join DS_DIST_NONE  (cost=0.00..6286.47 rows=172456 width=30)
```

```
                    Merge Cond: ("outer".listid = "inner".listid)
                    ->  XN Seq Scan on listing  (cost=0.00..1924.97 rows=192497
 width=14)
                    ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=24)
```

Dopo aver modificato la tabella dimensionale per utilizzare DISTSTYLE ALL, il piano di query per la stessa query mostra DS_DIST_ALL_NONE al posto di DS_BCAST_INNER. Inoltre, si verifica un significativo cambiamento in termini di costi per le fasi di combinazione. Il costo totale è `14142.59` confrontato con `3272334142.59` della query precedente.

```
->  XN Hash Join DS_DIST_ALL_NONE  (cost=112.50..14142.59 rows=170771 width=84)
        Hash Cond: ("outer".venueid = "inner".venueid)
        ->  XN Hash Join DS_DIST_ALL_NONE  (cost=109.98..10276.71 rows=172456 width=47)
              Hash Cond: ("outer".eventid = "inner".eventid)
              ->  XN Merge Join DS_DIST_NONE  (cost=0.00..6286.47 rows=172456 width=30)
                    Merge Cond: ("outer".listid = "inner".listid)
                    ->  XN Seq Scan on listing  (cost=0.00..1924.97 rows=192497
 width=14)
                    ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=24)
```

Esempio del piano di query

Questo esempio mostra come valutare un piano di query per scoprire le opportunità di ottimizzazione della distribuzione.

Esegui la seguente query con un comando EXPLAIN per produrre un piano di query.

```
explain
select lastname, catname, venuename, venuecity, venuestate, eventname,
month, sum(pricepaid) as buyercost, max(totalprice) as maxtotalprice
from category join event on category.catid = event.catid
join venue on venue.venueid = event.venueid
join sales on sales.eventid = event.eventid
join listing on sales.listid = listing.listid
join date on sales.dateid = date.dateid
join users on users.userid = sales.buyerid
group by lastname, catname, venuename, venuecity, venuestate, eventname, month
having sum(pricepaid)>9999
order by catname, buyercost desc;
```

Nel database TICKIT, SALES è una tabella dei fatti e LISTING è la sua dimensione più grande. Per collocare le tabelle, SALES viene distribuito su LISTID, che rappresenta la chiave esterna di LISTING, il quale viene distribuito sulla chiave primaria, LISTID. L'esempio seguente mostra il comando CREATE TABLE per SALES e LISTING.

```
create table sales(
 salesid integer not null,
 listid integer not null distkey,
 sellerid integer not null,
 buyerid integer not null,
 eventid integer not null encode mostly16,
 dateid smallint not null,
 qtysold smallint not null encode mostly8,
 pricepaid decimal(8,2) encode delta32k,
 commission decimal(8,2) encode delta32k,
 saletime timestamp,
 primary key(salesid),
 foreign key(listid) references listing(listid),
 foreign key(sellerid) references users(userid),
 foreign key(buyerid) references users(userid),
 foreign key(dateid) references date(dateid))
        sortkey(listid,sellerid);

create table listing(
 listid integer not null distkey sortkey,
 sellerid integer not null,
 eventid integer not null encode mostly16,
 dateid smallint not null,
 numtickets smallint not null encode mostly8,
 priceperticket decimal(8,2) encode bytedict,
 totalprice decimal(8,2) encode mostly32,
 listtime timestamp,
 primary key(listid),
 foreign key(sellerid) references users(userid),
 foreign key(eventid) references event(eventid),
 foreign key(dateid) references date(dateid));
```

Nel seguente piano di query, la fase Merge Join per la combinazione su SALES e LISTING mostra DS_DIST_NONE, che indica che non è necessaria alcuna ridistribuzione per la fase. Ad ogni modo, scorrendo il piano di query, le altre combinazioni interne mostrano DS_BCAST_INNER, il quale indica che la tabella interna viene trasmessa come parte dell'esecuzione della query. Dato che solo una

coppia di tabelle può essere collocata mediante la distribuzione della chiave, cinque tabelle devono essere ritrasmesse.

```
QUERY PLAN
XN Merge  (cost=1015345167117.54..1015345167544.46 rows=1000 width=103)
  Merge Key: category.catname, sum(sales.pricepaid)
  ->  XN Network  (cost=1015345167117.54..1015345167544.46 rows=170771 width=103)
        Send to leader
        ->  XN Sort  (cost=1015345167117.54..1015345167544.46 rows=170771 width=103)
              Sort Key: category.catname, sum(sales.pricepaid)
              ->  XN HashAggregate  (cost=15345150568.37..15345152276.08 rows=170771
 width=103)
                    Filter: (sum(pricepaid) > 9999.00)
                     ->  XN Hash Join DS_BCAST_INNER  (cost=742.08..15345146299.10
 rows=170771 width=103)
                           Hash Cond: ("outer".catid = "inner".catid)
                           ->  XN Hash Join DS_BCAST_INNER
 (cost=741.94..15342942456.61 rows=170771 width=97)
                                 Hash Cond: ("outer".dateid = "inner".dateid)
                                 ->  XN Hash Join DS_BCAST_INNER
 (cost=737.38..15269938609.81 rows=170766 width=90)
                                       Hash Cond: ("outer".buyerid = "inner".userid)
                                       ->  XN Hash Join DS_BCAST_INNER
 (cost=112.50..3272334142.59 rows=170771 width=84)
                                             Hash Cond: ("outer".venueid =
 "inner".venueid)
                                             ->  XN Hash Join DS_BCAST_INNER
 (cost=109.98..3167290276.71 rows=172456 width=47)
                                                   Hash Cond: ("outer".eventid =
 "inner".eventid)
                                                   ->  XN Merge Join DS_DIST_NONE
 (cost=0.00..6286.47 rows=172456 width=30)
                                                         Merge Cond: ("outer".listid =
 "inner".listid)
                                                         ->  XN Seq Scan on listing
 (cost=0.00..1924.97 rows=192497 width=14)
                                                         ->  XN Seq Scan on sales
 (cost=0.00..1724.56 rows=172456 width=24)
                                                   ->  XN Hash  (cost=87.98..87.98
 rows=8798 width=25)
                                                         ->  XN Seq Scan on event
 (cost=0.00..87.98 rows=8798 width=25)
```

```
                                          ->  XN Hash  (cost=2.02..2.02 rows=202
width=41)
                                                ->  XN Seq Scan on venue
(cost=0.00..2.02 rows=202 width=41)
                                    ->  XN Hash  (cost=499.90..499.90 rows=49990
width=14)
                                          ->  XN Seq Scan on users
(cost=0.00..499.90 rows=49990 width=14)
                              ->  XN Hash  (cost=3.65..3.65 rows=365 width=11)
                                    ->  XN Seq Scan on date  (cost=0.00..3.65
rows=365 width=11)
                         ->  XN Hash  (cost=0.11..0.11 rows=11 width=10)
                               ->  XN Seq Scan on category  (cost=0.00..0.11 rows=11
width=10)
```

Una soluzione è modificare le tabelle per avere DISTSTYLE ALL.

```
ALTER TABLE users ALTER DISTSTYLE ALL;
ALTER TABLE venue ALTER DISTSTYLE ALL;
ALTER TABLE category ALTER DISTSTYLE ALL;
ALTER TABLE date ALTER DISTSTYLE ALL;
ALTER TABLE event ALTER DISTSTYLE ALL;
```

Esegui la stessa query ancora con il comando EXPLAIN ed esamina il nuovo piano di query. Le combinazioni ora mostrano DS_DIST_ALL_NONE, il quale indica che non è necessaria alcuna ridistribuzione, dato che i dati sono stati distribuiti su tutti i nodi tramite DISTSTYLE ALL.

```
QUERY PLAN
XN Merge  (cost=1000000047117.54..1000000047544.46 rows=1000 width=103)
  Merge Key: category.catname, sum(sales.pricepaid)
  ->  XN Network  (cost=1000000047117.54..1000000047544.46 rows=170771 width=103)
        Send to leader
        ->  XN Sort  (cost=1000000047117.54..1000000047544.46 rows=170771 width=103)
              Sort Key: category.catname, sum(sales.pricepaid)
              ->  XN HashAggregate  (cost=30568.37..32276.08 rows=170771 width=103)
                    Filter: (sum(pricepaid) > 9999.00)
                    ->  XN Hash Join DS_DIST_ALL_NONE  (cost=742.08..26299.10
 rows=170771 width=103)
                          Hash Cond: ("outer".buyerid = "inner".userid)
                          ->  XN Hash Join DS_DIST_ALL_NONE  (cost=117.20..21831.99
 rows=170766 width=97)
                                Hash Cond: ("outer".dateid = "inner".dateid)
```

```
                                ->  XN Hash Join DS_DIST_ALL_NONE
(cost=112.64..17985.08 rows=170771 width=90)
                                      Hash Cond: ("outer".catid = "inner".catid)
                                      ->  XN Hash Join DS_DIST_ALL_NONE
(cost=112.50..14142.59 rows=170771 width=84)
                                            Hash Cond: ("outer".venueid =
"inner".venueid)
                                            ->  XN Hash Join DS_DIST_ALL_NONE
(cost=109.98..10276.71 rows=172456 width=47)
                                                  Hash Cond: ("outer".eventid =
"inner".eventid)
                                                  ->  XN Merge Join DS_DIST_NONE
(cost=0.00..6286.47 rows=172456 width=30)
                                                        Merge Cond: ("outer".listid =
"inner".listid)
                                                        ->  XN Seq Scan on listing
(cost=0.00..1924.97 rows=192497 width=14)
                                                        ->  XN Seq Scan on sales
(cost=0.00..1724.56 rows=172456 width=24)
                                                  ->  XN Hash  (cost=87.98..87.98
rows=8798 width=25)
                                                        ->  XN Seq Scan on event
(cost=0.00..87.98 rows=8798 width=25)
                                            ->  XN Hash  (cost=2.02..2.02 rows=202
width=41)
                                                  ->  XN Seq Scan on venue
(cost=0.00..2.02 rows=202 width=41)
                                      ->  XN Hash  (cost=0.11..0.11 rows=11 width=10)
                                            ->  XN Seq Scan on category
(cost=0.00..0.11 rows=11 width=10)
                                ->  XN Hash  (cost=3.65..3.65 rows=365 width=11)
                                      ->  XN Seq Scan on date  (cost=0.00..3.65
rows=365 width=11)
                          ->  XN Hash  (cost=499.90..499.90 rows=49990 width=14)
                                ->  XN Seq Scan on users  (cost=0.00..499.90 rows=49990
width=14)
```

Esempi di distribuzione

I seguenti esempi mostrano come i dati vengono distribuiti secondo le opzioni che hai definito nella dichiarazione CREATE TABLE.

Esempi DISTKEY

Guarda lo schema della tabella USERS nel database TICKIT. USERID è definito come la colonna SORTKEY e la colonna DISTKEY:

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'users';

    column     |          type          | encoding | distkey | sortkey
---------------+------------------------+----------+---------+---------
 userid        | integer                | none     | t       |       1
 username      | character(8)           | none     | f       |       0
 firstname     | character varying(30)  | text32k  | f       |       0

...
```

USERID è una buona scelta per la colonna di distribuzione su questa tabella. Se si esegue la query della vista di sistema SVV_DISKUSAGE, è possibile appurare che la tabella è stata distribuita in modo molto uniforme. I segmenti sono in base zero, quindi USERID è la colonna 0.

```
select slice, col, num_values as rows, minvalue, maxvalue
from svv_diskusage
where name='users' and col=0 and rows>0
order by slice, col;

slice| col | rows  | minvalue | maxvalue
-----+-----+-------+----------+----------
0    | 0   | 12496 | 4        | 49987
1    | 0   | 12498 | 1        | 49988
2    | 0   | 12497 | 2        | 49989
3    | 0   | 12499 | 3        | 49990
(4 rows)
```

La tabella contiene 49.990 righe. Le colonne delle righe (num_values) mostrano che ogni sezione contiene approssimativamente lo stesso numero di righe. Le colonne minvalue e maxvalue mostrano l'intervallo di valori su ogni sezione. Ogni sezione include quasi tutto l'intervallo di valori, quindi esiste una buona possibilità che ogni sezione partecipi all'esecuzione della query che filtra un intervallo di ID utente.

Questo esempio dimostra la distribuzione su un piccolo sistema di test. Il numero totale delle sezioni è solitamente più alto.

Se di solito esegui combinazioni o raggruppamenti mediante la colonna STATE, potresti scegliere di distribuire sulla colonna STATE. I seguenti esempi mostrano che se viene creata una nuova tabella con gli stessi dati della tabella USERS ma si imposta DISTKEY sulla colonna STATE, la distribuzione non sarà così ordinata. In questo caso, la distribuzione non è altrettanto uniforme. La sezione 0 (13.587 righe) contiene approssimativamente il 30% di righe in più rispetto alla sezione 3 (10.150 righe). In una tabella più grande, il totale della differenza di distribuzione potrebbe avere un impatto avverso sull'elaborazione di query.

```
create table userskey distkey(state) as select * from users;

select slice, col, num_values as rows, minvalue, maxvalue from svv_diskusage
where name = 'userskey' and col=0 and rows>0
order by slice, col;

slice | col | rows  | minvalue | maxvalue
------+-----+-------+----------+----------
    0 |   0 | 13587 |        5 |    49989
    1 |   0 | 11245 |        2 |    49990
    2 |   0 | 15008 |        1 |    49976
    3 |   0 | 10150 |        4 |    49986
(4 rows)
```

Esempio DISTSTYLE EVEN

Se crei una nuova tabella con gli stessi dati della tabella USERS ma imposti DISTSTYLE su EVEN, le righe saranno comunque distribuite in modo ordinato per le sezioni.

```
create table userseven diststyle even as
select * from users;

select slice, col, num_values as rows, minvalue, maxvalue from svv_diskusage
where name = 'userseven' and col=0 and rows>0
order by slice, col;

slice | col | rows  | minvalue | maxvalue
------+-----+-------+----------+----------
    0 |   0 | 12497 |        4 |    49990
    1 |   0 | 12498 |        8 |    49984
    2 |   0 | 12498 |        2 |    49988
    3 |   0 | 12497 |        1 |    49989
(4 rows)
```

Ad ogni modo, dato che la distribuzione non è basata su una colonna specifica, l'elaborazione di query può essere degradata, soprattutto se la tabella viene combinata su un'altra tabella. La mancanza di distribuzione su una colonna di combinazione, spesso influenza il tipo di operazione di combinazione che può essere eseguita in modo efficiente. Le operazioni di combinazione, aggregazione e raggruppamento sono ottimizzate quando entrambe le tabelle vengono distribuite e ordinate sulle loro rispettive colonne di combinazione.

Esempio DISTSTYLE ALL

Se crei una nuova tabella con gli stessi dati della tabella USERS ma imposti DISTSTYLE su ALL, tutte le righe saranno distribuite alla prima sezione di ogni nodo.

```
select slice, col, num_values as rows, minvalue, maxvalue from svv_diskusage
where name = 'usersall' and col=0 and rows > 0
order by slice, col;

slice | col | rows  | minvalue | maxvalue
------+-----+-------+----------+----------
    0 |   0 | 49990 |        4 |    49990
    2 |   0 | 49990 |        2 |    49990

(4 rows)
```

Utilizzo delle chiavi di ordinamento

> **Note**
>
> È consigliabile creare le tabelle con `SORTKEY AUTO`. In tal caso, Amazon Redshift utilizza l'ottimizzazione automatica della tabella per scegliere la chiave di ordinamento. Per ulteriori informazioni, consultare Utilizzo dell'ottimizzazione automatica delle tabelle. I dettagli sull'ordinamento sono descritti in dettaglio nella parte restante di questa sezione.

Quando si crea una tabella, è possibile definire una o più delle sue colonne come chiavi di ordinamento. Quando i dati vengono caricati inizialmente nella tabella vuota, le righe vengono archiviate sul disco in modo ordinato. Le informazioni sulle colonne della chiave di ordinamento vengono passate al pianificatore di query, che utilizza questa informazioni per costruire piani che tengano in considerazione il modo in cui sono ordinati i dati. Per ulteriori informazioni, consulta

CREATE TABLE. Per informazioni sulle best practice per la creazione di una chiave di ordinamento, consulta Scelta della migliore chiave di ordinamento.

L'ordinamento consente una gestione efficiente dei predicati a intervallo limitato. Amazon Redshift archivia i dati colonnari nei blocchi del disco da 1 MB. I valori minimo e massimo di ogni blocco vengono archiviati come parte dei metadati. Se una query utilizza un predicato a intervallo limitato, il processore di query può utilizzare i valori minimo e massimo per saltare rapidamente grandi numeri di blocchi durante la scansione della tabella. Ad esempio, si supponga che una tabella memorizzi cinque anni di dati ordinati per data e che una query specifichi un intervallo di un mese. In questo caso, puoi eliminare fino al 98% dei blocchi del disco dalla scansione. Se i dati non sono ordinati, dovranno essere scansionati più blocchi del disco (probabilmente tutti).

È possibile specificare la chiave di ordinamento composto o interlacciato. Una chiave di ordinamento composta è più efficiente quando i predicati di query utilizzano un prefisso, ovvero un sottoinsieme delle colonne della chiave di ordinamento in ordine. Una chiave di ordinamento interlacciato dà lo stesso peso a ogni colonna nella chiave di ordinamento, quindi i predicato di query possono utilizzare qualsiasi sottoinsieme di colonne che costituiscono la chiave di ordinamento, in qualsiasi ordine.

Per comprendere l'impatto sulle prestazioni della query della chiave di ordinamento scelta, utilizza il comando EXPLAIN. Per ulteriori informazioni, consultare Pianificazione di query e flusso di lavoro di esecuzione.

Per definire un tipo di ordinamento, utilizza la parola chiave INTERLEAVED o COMPOUND con le tue dichiarazioni CREATE TABLE o CREATE TABLE AS. L'impostazione predefinita è COMPOUND. COMPOUND è consigliata quando aggiorni regolarmente le tabelle con le operazioni INSERT, UPDATE o DELETE. Una chiave di ordinamento INTERLEAVED può utilizzare un massimo di otto colonne. A seconda delle dimensioni di dati e cluster, VACUUM REINDEX può impiegare molto più tempo rispetto a VACUUM FULL perché esegue un ulteriore passaggio per analizzare le chiavi di ordinamento interlacciate. L'operazione di ordinamento e unione può richiedere più tempo per le tabelle interlacciate perché l'ordinamento interlacciato potrebbe dover riordinare più righe rispetto a un ordinamento composto.

Per visualizzare le chiavi di ordinamento per una tabella, eseguire la query della vista di sistema SVV_TABLE_INFO .

Argomenti

- Ordinamento del layout dei dati multidimensionali (anteprima)
- Chiave di ordinamento composta
- Chiave di ordinamento interlacciato

Ordinamento del layout dei dati multidimensionali (anteprima)

Questa è una documentazione di pre-rilascio per l'ordinamento delle tabelle con layout di dati multidimensionali, disponibile nella versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

> **Note**
>
> Questa funzionalità è disponibile solo in un cluster o un gruppo di lavoro di anteprima. Per creare un cluster di anteprima, consulta Creazione di un cluster di anteprima nella Guida alla gestione di Amazon Redshift. Per creare un gruppo di lavoro di anteprima, consulta Creazione di un gruppo di lavoro di anteprima nella Guida alla gestione di Amazon Redshift.

Una chiave di ordinamento del layout di dati multidimensionali è una chiave di ordinamento di tipo AUTO basato su predicati ripetitivi presenti in un carico di lavoro. Se il carico di lavoro contiene predicati ripetitivi, Amazon Redshift può migliorare le prestazioni di scansione delle tabelle eseguendo la co-locazione delle righe di dati che soddisfano i predicati ripetitivi. Anziché archiviare i dati di una tabella in un rigoroso ordine di colonne, una chiave di ordinamento del layout dei dati multidimensionale memorizza i dati analizzando i predicati ripetitivi presenti in un carico di lavoro. In un carico di lavoro è possibile trovare più di un predicato ripetitivo. A seconda del carico di lavoro, questo tipo di chiave di ordinamento può migliorare le prestazioni di molti predicati. Amazon Redshift determina automaticamente se questo metodo della chiave di ordinamento deve essere utilizzato per le tabelle definite con una chiave di ordinamento `AUTO`.

Ad esempio, supponi di disporre di una tabella con i dati ordinati in base alle colonne. Potrebbe essere necessario esaminare molti blocchi di dati per determinare se soddisfano i predicati del carico di lavoro. Tuttavia, se i dati sono archiviati su disco nell'ordine dei predicati, è necessario scansionare un numero inferiore di blocchi per soddisfare la query. In questo caso è utile utilizzare una chiave di ordinamento del layout dei dati multidimensionali.

Per vedere se una query utilizza una chiave del layout dei dati multidimensionali, consulta la colonna `step_attribute` della vista SYS_QUERY_DETAIL. Quando il valore è `multi-dimensional`, per la query è stato utilizzato un layout di dati multidimensionali. Per vedere se una tabella definita

con la chiave di ordinamento AUTO utilizza un layout di dati multidimensionali, consulta la colonna `sortkey1` della vista SVV_TABLE_INFO. Quando il valore è `padb_internal_mddl_key_col`, per la chiave di ordinamento della tabella è stato utilizzato il layout dei dati multidimensionali.

Per evitare che Amazon Redshift utilizzi una chiave di ordinamento per il layout dei dati multidimensionali, scegli un'opzione di ordinamento della tabella diversa da `SORTKEY AUTO`. Per ulteriori informazioni sulle opzioni SORTKEY, consulta CREATE TABLE.

Chiave di ordinamento composta

Una chiave composta è costituita da tutte le colonne elencate nella definizione della chiave di ordinamento, nell'ordine in cui sono elencate. Una chiave di ordinamento composta è più utile quando un filtro di query applica delle condizioni, come filtri e combinazioni, che utilizzano un prefisso delle chiavi di ordinamento. I vantaggi in termini di prestazioni dell'ordinamento composto diminuiscono quando le query si basano solo su colonne di ordinamento secondarie, senza fare riferimento alle colonne primarie. COMPOUND è il tipo di ordinamento predefinito.

Le chiavi di ordinamento composte potrebbero velocizzare le combinazioni, le operazioni GROUP BY e ORDER BY e le funzioni di finestra che utilizzano PARTITION BY e ORDER BY. Ad esempio, un merge join, che è sempre più veloce di un hash join, è conveniente quando i dati vengono distribuiti e preordinati sulle colonne di combinazione. Le chiavi di ordinamento composte aiutano anche a migliorare la compressione.

Quando aggiungi righe a una tabella ordinata che già contiene dati, la regione non ordinata cresce; questo ha degli effetti importanti sulle prestazioni. L'effetto è maggiore quando la tabella utilizza ordinamento interlacciato, soprattutto quando le colonne di ordinamento includono dati che aumentano in maniera monotona, come le colonne di dati o timestamp. Esegui regolarmente un'operazione VACUUM, soprattutto dopo aver caricato grandi quantità di dati, per ordinare e analizzare nuovamente i dati. Per ulteriori informazioni, consulta Gestione delle dimensioni della regione non ordinata. Dopo il vacuum per riordinare i dati, è buona norma eseguire un comando ANALYZE per aggiornare i metadati statistici per il pianificatore di query. Per ulteriori informazioni, consultare Analisi delle tabelle.

Chiave di ordinamento interlacciato

Un ordinamento interlacciato dà lo stesso peso a ogni colonna, o sottoinsieme di colonne, nella chiave di ordinamento. Se più query utilizzano colonne diverse per i filtri, puoi spesso migliorare le prestazioni di quelle query tramite uno stile di ordinamento interlacciato. Quando una query utilizza

predicati restrittivi sulle colonne di ordinamento secondarie, l'ordinamento interlacciato migliora in modo significativo le prestazioni delle query rispetto all'ordinamento composto.

> **Important**
>
> Non utilizzare una chiave di ordinamento "interlacciato" su colonne con attributi che crescono in maniera monotona, come colonne di identità, date o timestamp.

I miglioramenti delle prestazioni che hai ottenuto implementando una chiave di ordinamento interlacciato, dovrebbero essere confrontati con i tempi di caricamento e di vacuum.

Gli ordinamenti interlacciati sono più efficaci con query molto selettive, che filtrano su una o più colonne di ordinamento nella clausola WHERE, ad esempio `select c_name from customer where c_region = 'ASIA'`. I benefici dell'ordinamento interlacciato aumenta con il numero delle colonne ordinate che vengono limitate.

Un ordinamento interlacciato è più efficace con le tabelle grandi. L'ordinamento viene applicato su ogni sezione. Di conseguenza, un ordinamento interlacciato è più efficace quando una tabella è abbastanza grande da richiedere più blocchi da 1 MB per sezione. Qui, il processore di query può saltare una percentuale significativa dei blocchi utilizzando predicati restrittivi. Per visualizzare il numero di blocchi che utilizza una tabella, eseguire la query della vista di sistema STV_BLOCKLIST.

Quando si esegue un ordinamento su una singola colonna, un ordinamento interlacciato fornisce prestazioni migliori rispetto a un ordinamento composto, nel caso in cui i valori della colonna abbiano un prefisso lungo in comune. Ad esempio, di solito le URL cominciano con "http://www". Le chiavi di ordinamento composto utilizzano un numero limitati di caratteri dal prefisso, che comporta molte duplicazioni delle chiavi. Gli ordinamenti interlacciati utilizzano uno schema di compressione interno per i valori della mappa di zona, i quali gli consentono di discriminare meglio i valori delle colonne che dispongono di un prefisso lungo in comune.

Quando si migrano cluster con provisioning di Amazon Redshift ad Amazon Redshift Serverless, Redshift converte le tabelle con chiavi di ordinamento interlacciate e DISTSTYLE KEY in chiavi di ordinamento composte. DISTSTYLE non cambia. Per ulteriori informazioni sugli stili di distribuzione, consulta Utilizzo degli stili di distribuzione dati.

VACUUM REINDEX

Quando aggiungi righe a una tabella ordinata che già contiene dati, le prestazioni possono diminuire nel tempo. Questa diminuzione riguarda sia gli ordinamenti composti che quelli interlacciati, ma ha

degli effetti più significativi sulle tabelle interlacciate. Un VACUUM ripristina l'ordine, ma l'operazione può impiegare più tempo per le tabelle interlacciate, perché l'unione di nuovi dati interlacciati può causare modifiche ad ogni blocco di dati.

Quando le tabelle vengono caricate inizialmente, Amazon Redshift analizza la distribuzione dei valori nelle colonne della chiave di ordinamento e utilizza queste informazioni per un ordinamento interlacciato ottimale delle colonne della chiave di ordinamento. Quando una tabella cresce, la distribuzione dei valori nelle colonne della chiave di ordinamento può cambiare o differenziarsi, in particolar modo con le colonne timestamp o di dati. Se la differenza diventa troppo ampia, le prestazioni potrebbero risentirne. Per analizzare nuovamente le chiavi di ordinamento e per ripristinare le prestazioni, esegui il comando VACUUM con la parola chiave REINDEX. Dato che è necessaria un'ulteriore analisi da passare sui dati, VACUUM REINDEX può impiegare più tempo rispetto al VACUUM standard per le tabelle interlacciate. Per visualizzare le informazioni sulla differenza della distribuzione della chiave e sull'ultimo tempo di reindicizzazione, eseguire una query della vista di sistema SVV_INTERLEAVED_COLUMNS.

Per ulteriori informazioni su come determinare la frequenza dell'esecuzione di VACUUM e quando eseguire VACUUM REINDEX, consultare Decisione sulla reindicizzazione.

Definizione di limitazioni delle tabelle

In modo univoco, le limitazioni della chiave esterna e della chiave primaria sono solo a livello informativo e non vengono applicate da Amazon Redshift quando si popola una tabella. Ad esempio, se si inseriscono dati in una tabella con dipendenze, l'inserimento può avere esito positivo anche se viola il vincolo. Tuttavia, la chiave esterna e quella primaria vengono utilizzate come suggerimenti di pianificazione. Queste dovrebbero essere dichiarate se il processo ETL o altri processi nell'applicazione applicano la loro integrità.

Ad esempio, il pianificatore di query utilizza chiavi primarie ed esterne in determinati calcoli statistici. Lo fa per dedurre l'unicità e le relazioni referenziali che influiscono sulle tecniche di decorrelazione delle query secondarie. In questo modo, può ordinare un gran numero di join ed eliminare i join ridondanti.

Il pianificatore utilizza queste relazioni della chiave, ma presuppone che tutte le chiavi nelle tabelle Amazon Redshift vengano validate appena caricate. Se la tua applicazione permette chiavi primarie o esterne non valide, alcune query potrebbero restituire risultati sbagliati. Ad esempio, una query SELECT DISTINCT potrebbe restituire righe doppie se la chiave primaria non è univoca. Non definire i limiti della chiave per le tue tabelle se hai dei dubbi sulla loro validità. D'altra parte, si dovrebbe

sempre dichiarare la chiave primaria e quella esterna, oltre che i limiti di univocità se si sa che sono validi.

Amazon Redshift applica i limiti della colonna NOT NULL.

Per ulteriori informazioni sui vincoli di tabella, consultare CREATE TABLE. Per informazioni su come eliminare una tabella con dipendenze, consultare DROP TABLE.

Caricamento dei dati

Argomenti

Il modo più efficiente per caricare una tabella è il comando COPY. Inoltre, è possibile aggiungere dati alle tabelle usando i comandi INSERT, sebbene sia molto meno efficiente rispetto all'utilizzo di COPY. Il comando COPY è in grado di leggere da più file di dati o più flussi di dati contemporaneamente. Amazon Redshift assegna il carico di lavoro ai nodi del cluster ed esegue le operazioni di caricamento in parallelo, incluso l'ordinamento delle righe e la distribuzione dei dati tra le sezioni dei nodi.

> **Note**
>
> Le tabelle esterne di Amazon Redshift Spectrum sono di sola lettura. Non puoi utilizzare COPY o INSERT in una tabella esterna.

Per accedere ai dati su altre AWS risorse, il cluster deve disporre dell'autorizzazione ad accedere a tali risorse e ad eseguire le azioni necessarie per accedere ai dati. È possibile utilizzare AWS Identity and Access Management (IAM) per limitare l'accesso degli utenti alle risorse e ai dati del cluster.

Dopo il caricamento iniziale dei dati, se aggiungi, modifichi o elimini una grande quantità di dati, devi eseguire il follow-up tramite un comando VACUUM per riorganizzare i dati e recuperare spazio dopo le eliminazioni. Devi anche eseguire un comando ANALYZE per aggiornare le statistiche della tabella.

Questa sezione descrive come caricare i dati e risolvere i problemi relativi ai caricamenti dei dati e presenta le best practice per il caricamento dei dati.

Utilizzo di un comando COPY per il caricamento dei dati

Argomenti

Il comando COPY sfrutta l'architettura MPP (Massively Parallel Processing) di Amazon Redshift per leggere e caricare dati in parallelo da file su Amazon S3, da una tabella DynamoDB o da output di testo da uno o più host remoti.

> **Note**
>
> Consigliamo l'uso del comando COPY per caricare grandi quantità di dati. L'utilizzo di singole istruzioni INSERT per popolare una tabella potrebbe essere eccessivamente lento. In alternativa, se i dati sono già presenti in altre tabelle di database Amazon Redshift, utilizzare INSERT INTO... SELECT o CREATE TABLE AS per migliorare le prestazioni. Per informazioni, consulta INSERT or CREATE TABLE AS.

Per caricare dati da un'altra AWS risorsa, il cluster deve disporre dell'autorizzazione per accedere alla risorsa ed eseguire le azioni necessarie.

Per concedere o revocare il privilegio per caricare dati in una tabella utilizzando un comando COPY, concedi o revoca il privilegio INSERT.

I dati devono essere nel formato corretto per il caricamento nella tabella Amazon Redshift. Questa sezione presenta le linee guida per preparare e verificare i dati prima del caricamento e per convalidare un'istruzione COPY prima di eseguirla.

Per proteggere le informazioni nei file, è possibile eseguire la crittografia dei file di dati prima di caricarli nel bucket Amazon S3; COPY eseguirà la decrittografia dei dati mentre esegue il caricamento. Inoltre, è possibile limitare l'accesso ai dati del caricamento fornendo credenziali di sicurezza temporanee agli utenti. Le credenziali di sicurezza temporanee offrono maggiore sicurezza perché hanno una durata breve e non possono essere riutilizzate dopo la loro scadenza.

Amazon Redshift dispone di funzionalità integrate per COPY per caricare rapidamente dati non compressi e delimitati. È possibile comprimere i file tramite gzip, lzop o bzip2 per risparmiare tempo nel caricamento dei file.

Se nella query COPY si trovano le seguenti parole chiave, la suddivisione automatica dei dati non compressi non è supportata: ESCAPE, REMOVEQUOTES e FIXEDWIDTH. Ma la parola chiave CSV è supportata.

Per garantire la sicurezza dei dati in transito all'interno del AWS cloud, Amazon Redshift utilizza SSL con accelerazione hardware per comunicare con Amazon S3 o Amazon DynamoDB per le operazioni di COPY, UNLOAD, backup e ripristino.

Quando si carica la tabella direttamente da una tabella di Amazon DynamoDB, si ha la possibilità di controllare la quantità di throughput assegnato di Amazon DynamoDB che viene utilizzato.

Facoltativamente, è possibile consentire a COPY di analizzare i dati di input e applicare automaticamente codifiche di compressione ottimali alla tabella come parte del processo di caricamento.

Credenziali e autorizzazioni di accesso

Per caricare o scaricare dati utilizzando un'altra AWS risorsa, come Amazon S3, Amazon DynamoDB, Amazon EMR o Amazon EC2, il cluster deve disporre dell'autorizzazione per accedere alla risorsa ed eseguire le azioni necessarie per accedere ai dati. Ad esempio, per caricare i dati da Amazon S3, COPY deve avere accesso LIST al bucket e accesso GET per gli oggetti del bucket.

Per ottenere l'autorizzazione per accedere a una risorsa, il cluster deve essere autenticato. È possibile scegliere il controllo degli accessi basato su ruoli o il controllo degli accessi basato su

chiave. Questa sezione presenterà una panoramica dei due metodi. Per esempi e dettagli completi, consultare Autorizzazioni per accedere ad altre risorse AWS.

Controllo degli accessi basato sui ruoli

Con il controllo degli accessi basato su ruoli, il cluster assume in modo temporaneo e automatico un ruolo AWS Identity and Access Management (IAM). Quindi, in base alle autorizzazioni concesse al ruolo, il cluster può accedere alle risorse richieste. AWS

Ti consigliamo di utilizzare il controllo degli accessi basato sui ruoli perché fornisce un controllo più sicuro e dettagliato dell'accesso alle AWS risorse e ai dati sensibili degli utenti, oltre a salvaguardare le credenziali. AWS

Per utilizzare il controllo degli accessi basato su ruoli, è necessario creare prima un ruolo IAM utilizzando il tipo di ruolo di servizio Amazon Redshift, quindi collegare il ruolo al cluster. Il ruolo deve avere, come minimo, le autorizzazioni elencate in Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY. Per i passaggi per creare un ruolo IAM e collegarlo al cluster, consulta Creazione di un ruolo IAM per consentire al cluster Amazon Redshift di accedere ai AWS servizi nella Guida alla gestione di Amazon Redshift.

È possibile aggiungere un ruolo a un cluster o visualizzare i ruoli associati a un cluster utilizzando la Console di gestione, la CLI o l'API di Amazon Redshift. Per ulteriori informazioni, consulta Autorizzazione delle operazioni COPY e UNLOAD mediante ruoli IAM nella Guida alla gestione di Amazon Redshift.

Quando crei un ruolo IAM, IAM restituisce un Amazon Resource Name (ARN) per il ruolo. Per eseguire il comando COPY tramite il ruolo IAM, fornisci l'ARN del ruolo tramite il parametro IAM_ROLE o il parametro CREDENTIALS.

Il seguente esempio di comando COPY utilizza il parametro IAM_ROLE con il ruolo `MyRedshiftRole` per l'autenticazione.

```
copy customer from 's3://mybucket/mydata'
iam_role 'arn:aws:iam::12345678901:role/MyRedshiftRole';
```

L' AWS utente deve disporre almeno delle autorizzazioni elencate in. Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY

Controllo degli accessi basato su chiave

Con il controllo degli accessi basato su chiavi, fornisci l'ID della chiave di accesso e la chiave di accesso segreta per un utente autorizzato ad accedere alle AWS risorse che contengono i dati.

> **Note**
>
> Consigliamo vivamente di utilizzare un ruolo IAM per l'autenticazione invece di fornire un ID chiave di accesso in chiaro e una chiave di accesso segreta. Se scegli il controllo degli accessi basato su chiavi, non utilizzare mai le credenziali del tuo AWS account (root). Creare sempre un utente IAM e fornire l'ID della chiave di accesso e la chiave di accesso segreta di quell'utente. Per la procedura per creare un utente IAM, consultare Creazione di un utente IAM nell'account AWS.

Preparazione dei dati di input

Se i dati di input non sono compatibili con le colonne della tabella che li riceveranno, il comando COPY fallirà.

Attieniti alle seguenti linee guida per garantire che i dati di input siano validi:

- I dati possono contenere solo fino a quattro byte di caratteri con codice UTF-8.
- Verifica che le stringhe CHAR e VARCHAR non siano più lunghe delle colonne corrispondenti. Le stringhe VARCHAR sono misurate in byte, non in caratteri, quindi, ad esempio, una stringa di quattro caratteri di caratteri cinesi che occupano quattro byte ciascuno richiede una colonna VARCHAR(16).
- I caratteri multibyte possono essere utilizzati solo con colonne VARCHAR. Verifica che i caratteri multibyte non siano più lunghi di quattro byte.
- Verifica che i dati per le colonne CHAR contengano solo caratteri a byte singolo.
- Non includere caratteri speciali o sintassi per indicare l'ultimo campo in un record. Questo campo può essere un delimitatore.
- Se i dati includono terminatori null, indicati anche come NUL (UTF-8 0000) o zero binario (0x000), è possibile caricare questi caratteri come NULLS nelle colonne CHAR o VARCHAR tramite l'opzione NULL AS nel comando COPY: `null as '\0'` o `null as '\000'` . Se non utilizzi NULL AS, i terminatori null causeranno il fallimento del COPY.

- Se le stringhe contengono caratteri speciali, come delimitatori e nuove righe incorporate, utilizza l'opzione ESCAPE con il comando COPY.
- Verificare che tutte le virgolette singole e doppie siano abbinate correttamente.
- Verifica che le stringhe a virgola mobile siano in formato a virgola mobile standard, ad esempio 12.123 o in formato esponenziale, ad esempio 1.0E4.
- Verifica che tutte le stringhe timestamp e data seguano le specifiche per Stringhe DATEFORMAT e TIMEFORMAT. Il formato timestamp predefinito è AAAA-MM-GG hh:mm:ss e il formato data predefinito è AAAA-MM-GG.
- Per ulteriori informazioni sui confini e sulle limitazioni sui singoli tipi di dati, consultare Tipi di dati. Per informazioni sugli errori di carattere multibyte, consultare Errori di caricamento di caratteri multibyte

Caricamento di dati da Amazon S3

Argomenti

- Caricamento dei dati da file compressi e non compressi
- Caricamento di file in Amazon S3
- Utilizzo del comando COPY per il caricamento da Amazon S3

Il comando COPY sfrutta l'architettura MPP (Massively Parallel Processing) di Amazon Redshift per leggere e caricare dati in parallelo da uno o più file in un bucket Amazon S3. È possibile ottenere il massimo dall'elaborazione parallela suddividendo i dati in più file, nei casi in cui i file sono compressi. Ci sono eccezioni a questa regola. Queste sono descritte in dettaglio in Caricamento dei file di dati. È anche possibile ottenere il massimo dall'elaborazione parallela impostando le chiavi di distribuzione nelle tabelle. Per ulteriori informazioni sulle chiavi di distribuzione, consulta Utilizzo degli stili di distribuzione dati.

I dati vengono caricati nella tabella di destinazione, una riga per riga. I campi nel file di dati corrispondono alle colonne della tabella in ordine, da sinistra a destra. I campi nei file di dati possono essere a larghezza fissa o delimitati da caratteri; il delimitatore predefinito è una barra verticale (|). Per impostazione predefinita, vengono caricate tutte le colonne della tabella, ma è possibile definire facoltativamente un elenco di colonne separato da virgole. Se una colonna della tabella non è inclusa nell'elenco delle colonne specificato nel comando COPY, viene caricata con un valore predefinito. Per ulteriori informazioni, consulta Caricamento dei valori delle colonne predefiniti.

Caricamento dei dati da file compressi e non compressi

Quando carichi dati compressi, ti consigliamo di suddividere i dati per ogni tabella in più file. Quando carichi dati non compressi e delimitati, il comando COPY utilizza l'elaborazione massivamente parallela (MPP) e gli intervalli di scansione per caricare dati da file di grandi dimensioni in un bucket Amazon S3.

Caricamento dei dati da più file compressi

Nei casi in cui sono stati compressi dati, ti consigliamo di suddividerli per ogni tabella in più file. Il comando COPY può caricare dati da più file in parallelo. È possibile caricare più file specificando un prefisso comune o chiave di prefisso, per l'insieme o elencando in modo esplicito i file in un file manifest.

Dividi i dati in file in modo che il numero di file sia un multiplo del numero di sezioni nel cluster. In questo modo Amazon Redshift può dividere i dati in modo uniforme tra le sezioni. Il numero di sezioni per nodo dipende dalla dimensione dei nodi del cluster. Ad esempio, ogni nodo di calcolo dc2.large ha due slice e ogni nodo di calcolo dc2.8xlarge ha 16 slice. Per ulteriori informazioni sul numero di sezioni per ogni dimensione di nodo, consulta Informazioni su cluster e nodi nella Guida alla gestione di Amazon Redshift.

Tutti i nodi partecipano all'esecuzione di query parallele, lavorando sui dati di calcolo distribuiti più uniformemente possibile all'interno delle sezioni. Se disponi di un cluster con due nodi dc2.large, potresti suddividere i dati in quattro file o più di quattro. Amazon Redshift non prende in considerazione la dimensione di file nella suddivisione del carico di lavoro. Pertanto, è necessario assicurarsi che i file abbiano approssimativamente le stesse dimensioni, da 1 MB a 1 GB dopo la compressione.

Per utilizzare i prefissi degli oggetti per identificare i file di caricamento, denomina ciascun file con un prefisso comune. Ad esempio, il file `venue.txt` può essere diviso in quattro file, come segue.

```
venue.txt.1
venue.txt.2
venue.txt.3
venue.txt.4
```

Se collochi più file in una cartella nel bucket, specificando il nome della cartella come prefisso, COPY carica tutti i file nella cartella. Se elenchi in modo esplicito i file da caricare tramite un file manifest, i file possono risiedere in diversi bucket o diverse cartelle.

Per ulteriori informazioni sui file manifest, consulta Example: COPY from Amazon S3 using a manifest.

Caricamento di dati da file delimitati e non compressi

Quando carichi dati delimitati e non compressi, il comando COPY utilizza l'architettura MPP (Massively Parallel Processing) di Amazon Redshift. Amazon Redshift utilizza automaticamente le sezioni che lavorano in parallelo per caricare intervalli di dati da un file di grandi dimensioni in un bucket Amazon S3. Il file deve essere delimitato per il caricamento parallelo. Ad esempio, tubo delimitato. Il caricamento automatico dei dati in parallelo con il comando COPY non è disponibile per i file CSV. È possibile utilizzare l'elaborazione parallela impostando le chiavi di distribuzione nelle tabelle. Per ulteriori informazioni sulle chiavi di distribuzione, consulta Utilizzo degli stili di distribuzione dati.

Il caricamento automatico dei dati in parallelo non è supportato se la query COPY include una delle parole chiave ESCAPE, REMOVEQUOTES e FIXEDWIDTH.

I dati dai file vengono caricati nella tabella di destinazione, una riga per riga. I campi nel file di dati corrispondono alle colonne della tabella in ordine, da sinistra a destra. I campi nei file di dati possono essere a larghezza fissa o delimitati da caratteri; il delimitatore predefinito è una barra verticale (|). Per impostazione predefinita, vengono caricate tutte le colonne della tabella, ma è possibile definire facoltativamente un elenco di colonne separato da virgole. Se una colonna della tabella non è inclusa nell'elenco delle colonne specificato nel comando COPY, viene caricata con un valore predefinito. Per ulteriori informazioni, consulta Caricamento dei valori delle colonne predefiniti.

Segui questo procedimento generale per caricare dati da Amazon S3, quando i dati non sono compressi e delimitati:

1. Caricare i file su Amazon S3.
2. Esegui un comando COPY per caricare la tabella.
3. Verifica i dati siano stati caricati correttamente.

Per esempi di comandi COPY, consultare Esempi di COPY. Per informazioni sui dati caricati in Amazon Redshift, consultare le tabelle di sistema STL_LOAD_COMMITS e STL_LOAD_ERRORS.

Per ulteriori informazioni sui nodi e sulle sezioni contenute in ciascun nodo, consulta Informazioni su cluster e nodi nella Guida alla gestione di Amazon Redshift.

Caricamento di file in Amazon S3

Argomenti

Ci sono un paio di approcci da adottare quando si caricano file di testo su Amazon S3:

- Se disponi di file compressi, ti consigliamo di dividere file di grandi dimensioni per beneficiare dall'elaborazione parallela in Amazon Redshift.
- D'altra parte, COPY divide automaticamente dati di file di grandi dimensioni, non compressi e delimitati da testo per facilitare il parallelismo e distribuire efficacemente i dati da file di grandi dimensioni.

Creare un bucket Amazon S3 per contenere i file di dati, quindi caricare i file di dati nel bucket. Per informazioni sulla creazione di bucket e sul caricamento di file, consultare Utilizzo dei bucket Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

> **Important**
>
> Il bucket Amazon S3 che contiene i file di dati deve essere creato nella stessa regione AWS del cluster a meno che non si utilizzi l'opzione REGION per specificare la regione in cui si trova il bucket Amazon S3.

Assicurati che gli intervalli IP S3 siano aggiunti all'elenco di indirizzi consentiti. Per ulteriori informazioni sugli intervalli IP S3 richiesti, consulta Isolamento di rete.

È possibile creare un bucket Amazon S3 in una regione specifica selezionando la regione quando si crea il bucket con la console Amazon S3 oppure specificando un endpoint quando si crea il bucket con la CLI o l'API di Amazon S3.

In seguito al caricamento dei dati, verificare che in Amazon S3 siano presenti i file corretti.

Gestione della coerenza dei dati

Amazon S3 offre una forte read-after-write coerenza per le operazioni COPY, UNLOAD, INSERT (tabella esterna), CREATE EXTERNAL TABLE AS e Amazon Redshift Spectrum sui bucket Amazon S3 in tutte le regioni. AWS Inoltre, le operazioni di lettura su Amazon S3 Select, gli elenchi di controllo accessi (ACL) di Amazon S3, i tag oggetto Amazon S3 e i metadati degli oggetti (ad esempio oggetto HEAD) sono fortemente consistenti. Per informazioni più dettagliate sulla coerenza dei dati, consultare Modello di consistenza dei dati di Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Caricamento di dati crittografati su Amazon S3

Amazon S3 supporta sia la crittografia lato server sia la crittografia lato client. In questo argomento vengono discusse le differenze tra la crittografia lato server e quella lato client e vengono descritte le fasi per utilizzare la crittografia lato client con Amazon Redshift. La crittografia lato server è trasparente per Amazon Redshift.

Crittografia lato server

La crittografia lato server è una crittografia dei dati a riposo, vale a dire che Amazon S3 consente di crittografare i dati durante il caricamento e ne esegue la decrittografia quando avviene l'accesso. Quando si esegue il caricamento delle tabelle utilizzando un comando COPY, non esiste differenza nel modo in cui si esegue il caricamento da oggetti non crittografati o crittografati lato server in Amazon S3. Per ulteriori informazioni sulla crittografia lato server, consultare Utilizzo della crittografia lato server nella Guida per l'utente di Amazon Simple Storage Service.

Crittografia lato client

Nella crittografia lato client, l'applicazione client gestisce la crittografia dei dati, le chiavi di crittografia e gli strumenti correlati. È possibile caricare i dati in un bucket Amazon S3 utilizzando la crittografia lato client, quindi caricare i dati utilizzando il comando COPY con l'opzione ENCRYPTED e una chiave di crittografia privata per fornire maggiore sicurezza.

Crittografi i dati utilizzando la crittografia envelope. Con la crittografia envelope, l'applicazione gestisce tutta la crittografia in modo esclusivo. Le tue chiavi di crittografia private e i tuoi dati non crittografati non vengono mai inviati a AWS, quindi è molto importante gestire in modo sicuro le chiavi di crittografia. Se perdi le chiavi di crittografia, non sarai in grado di decrittografare i dati e non potrai recuperarle da. AWS La crittografia envelope combina le prestazioni della crittografia simmetrica rapida, mantenendo al contempo la maggiore sicurezza fornita dalla gestione delle chiavi con chiavi asimmetriche. Una chiave one-time-use simmetrica (la chiave simmetrica dell'involucro)

viene generata dal client di crittografia Amazon S3 per crittografare i dati, quindi tale chiave viene crittografata dalla chiave principale e archiviata insieme ai dati in Amazon S3. Quando Amazon Redshift accede ai dati durante un caricamento, la chiave simmetrica crittografata viene recuperata e decrittografata con la chiave reale, quindi i dati vengono decrittografati.

Per lavorare con i dati crittografati lato client di Amazon S3 in Amazon Redshift, seguire la procedura descritta in Protezione dei dati con la crittografia lato client nella Guida per l'utente di Amazon Simple Storage Service, con i requisiti aggiuntivi utilizzati:

- Crittografia simmetrica: la classe di AWS SDK per Java `AmazonS3EncryptionClient` utilizza la crittografia envelope, descritta in precedenza, che si basa sulla crittografia della chiave simmetrica. Utilizzare questa classe per creare un client Amazon S3 per caricare dati crittografati lato client.
- Una chiave simmetrica root AES a 256 bit: una chiave root crittografa la chiave envelope. Trasferisci la chiave root all'istanza della chiave `AmazonS3EncryptionClient`. Salvare questa chiave, perché sarà necessaria per copiare i dati in Amazon Redshift.
- Metadati oggetto per archiviare la chiave envelope crittografata: per impostazione predefinita, Amazon S3 archivia la chiave envelope come metadati degli oggetti per la classe `AmazonS3EncryptionClient`. La chiave envelope crittografata che viene archiviata come metadati degli oggetti viene utilizzata durante il processo di decrittografia.

> **Note**
>
> Se si ottiene un messaggio indicante un errore di crittografia quando si usa l'API di crittografia per la prima volta, la propria versione del JDK potrebbe avere un file di policy per l'ambito di Java Cryptography Extension (JCE) che limita la lunghezza massima della chiave per le trasformazioni di crittografia e decrittografia a 128 bit. Per informazioni su come risolvere questo problema, consulta Specificare la crittografia lato client utilizzando l'SDK for AWS Java nella Guida per l'utente di Amazon Simple Storage Service.

Per informazioni sul caricamento di file crittografati lato client nelle tabelle Amazon Redshift con il comando COPY, consultare Caricamento di file di dati crittografati da Amazon S3.

Esempio: caricamento di dati crittografati lato client

Per un esempio di come utilizzare l' AWS SDK for Java per caricare dati crittografati lato client, consulta Protezione dei dati utilizzando la crittografia lato client nella Guida per l'utente di Amazon Simple Storage Service.

La seconda opzione illustra le scelte da effettuare durante la crittografia lato client in modo che i dati possano essere caricati in Amazon Redshift. Nello specifico, l'esempio illustra l'uso dei metadati degli oggetti per archiviare la chiave envelope crittografata e l'uso di una chiave simmetrica root AES a 256 bit.

Questo esempio fornisce un codice di esempio che utilizza l' AWS SDK for Java per creare una chiave radice simmetrica AES a 256 bit e salvarla in un file. Successivamente, nell'esempio viene caricato un oggetto in Amazon S3 utilizzando un client di crittografia S3 che prima crittografa i dati di esempio sul lato client. Nell'esempio si scarica anche l'oggetto e si verifica che i dati corrispondano.

Verifica della presenza dei file corretti nel bucket

Dopo aver caricato i file nel bucket Amazon S3, consigliamo di elencare i contenuti del bucket per verificare che siano presenti tutti i file corretti e che non siano presenti file indesiderati. Ad esempio, se il bucket `mybucket` conserva un file denominato `venue.txt.back`, quel file verrà caricato, forse involontariamente, dal seguente comando:

```
copy venue from 's3://mybucket/venue' … ;
```

Se desideri controllare specificamente quali file sono caricati, è possibile utilizzare un file manifest per elencare in modo esplicito i file di dati. Per ulteriori informazioni sull'utilizzo di un file manifest, consultare l'copy_from_s3_manifest_fileopzione per il comando COPY e Example: COPY from Amazon S3 using a manifest negli esempi COPY.

Per ulteriori informazioni sull'elenco dei contenuti del bucket, consultare Elenco delle chiavi degli oggetti nella Guida per gli sviluppatori di Amazon S3.

Utilizzo del comando COPY per il caricamento da Amazon S3

Argomenti

- Utilizzo di un manifest per specificare i fili di dati
- Caricamento di file di dati compressi da Amazon S3
- Caricamento di dati a larghezza fissa da Amazon S3
- Caricamento di dati multibyte da Amazon S3
- Caricamento di file di dati crittografati da Amazon S3

Utilizzare il comando COPY per caricare una tabella in parallelo dai file di dati in Amazon S3. È possibile specificare i file da caricare tramite il prefisso di un oggetto Amazon S3 o un file manifest.

La sintassi per specificare un file caricare tramite un prefisso è la seguente.

```
copy <table_name> from 's3://<bucket_name>/<object_prefix>'
authorization;
```

Il file manifest è un file in formato JSON che elenca i file di dati da caricare. La sintassi per specificare un file caricare tramite un file manifest è la seguente.

```
copy <table_name> from 's3://<bucket_name>/<manifest_file>'
authorization
manifest;
```

La tabella da caricare deve esistere già nel database. Per ulteriori informazioni sulla creazione di una tabella, consultare CREATE TABLE nel riferimento SQL.

I valori per l'autorizzazione forniscono l' AWS autorizzazione necessaria al cluster per accedere agli oggetti Amazon S3. Per informazioni sulle autorizzazioni richieste, consultare Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY. Il metodo preferito per l'autenticazione è specificare il parametro IAM_ROLE e fornire l'Amazon Resource Name (ARN) per un ruolo IAM con le autorizzazioni necessarie. Per ulteriori informazioni, consultare . Controllo degli accessi basato sui ruoli .

Per effettuare l'autenticazione utilizzando il parametro IAM_ROLE, sostituisci *<aws-account-id>* e *<role-name>* come mostrato nella sintassi seguente.

```
IAM_ROLE 'arn:aws:iam::<aws-account-id>:role/<role-name>'
```

L'esempio seguente mostra l'autenticazione utilizzando un ruolo IAM.

```
copy customer
from 's3://mybucket/mydata'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Per ulteriori informazioni su altre opzioni di autorizzazione, consulta Parametri di autorizzazione

Se desideri convalidare i dati senza caricare effettivamente la tabella, utilizza l'opzione NOLOAD con il comando COPY.

L'esempio seguente mostra le prime righe di dati delimitati da barra verticale in un file denominato `venue.txt`.

```
1|Toyota Park|Bridgeview|IL|0
2|Columbus Crew Stadium|Columbus|OH|0
3|RFK Stadium|Washington|DC|0
```

Prima di caricare il file in Amazon S3, dividere il file in più file in modo che il comando COPY possa caricarlo utilizzando l'elaborazione parallela. Il numero di file deve essere un multiplo del numero di sezioni nel cluster. Suddividi i file di dati di caricamento di modo che siano all'incirca della stessa dimensione, tra 1 MB e 1 GB dopo la compressione. Per ulteriori informazioni, consulta Caricamento dei dati da file compressi e non compressi.

Ad esempio, il file `venue.txt` può essere diviso in quattro file, come segue:

```
venue.txt.1
venue.txt.2
venue.txt.3
venue.txt.4
```

Il comando seguente COPY carica la tabella VENUE utilizzando i dati delimitati da barra verticale nei file di dati con il prefisso “venue” nel bucket Amazon S3 `mybucket`.

> **Note**
>
> Il bucket Amazon S3 `mybucket` nei seguenti esempi non esiste. Per i comandi COPY di esempio che utilizzano dati reali in un bucket Amazon S3 esistente, consultare Caricamento dei dati di esempio.

```
copy venue from 's3://mybucket/venue'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|';
```

Se non esiste alcun oggetto Amazon S3 con prefisso della chiave 'venue', il caricamento non riesce.

Utilizzo di un manifest per specificare i fili di dati

Puoi utilizzare un manifesto per assicurare che per un caricamento di dati il comando COPY carichi tutti i file richiesti e solo quelli. È possibile utilizzare un manifest per caricare file da diversi bucket o file che non condividono lo stesso prefisso. Anziché fornire il percorso di un oggetto per il comando COPY, fornisci il nome per un file di testo in formato JSON che elenca in modo esplicito i file da

caricare. L'URL nel manifest deve specificare il nome bucket e il percorso completo dell'oggetto per il file, non solo un prefisso.

Per ulteriori informazioni sui file manifest, consultare l'esempio di comando COPY Utilizzo di un manifest per specificare i file di dati.

L'esempio seguente mostra il JSON per il caricamento dei file da diversi bucket e con nomi di file che iniziano con stamp di data.

```
{
  "entries": [
    {"url":"s3://mybucket-alpha/2013-10-04-custdata", "mandatory":true},
    {"url":"s3://mybucket-alpha/2013-10-05-custdata", "mandatory":true},
    {"url":"s3://mybucket-beta/2013-10-04-custdata", "mandatory":true},
    {"url":"s3://mybucket-beta/2013-10-05-custdata", "mandatory":true}
  ]
}
```

Il flag `mandatory` facoltativo specifica se COPY deve restituire un errore qualora il file non venga trovato. Il valore predefinito di `mandatory` è `false`. Indipendentemente da eventuali impostazioni obbligatorie, COPY terminerà se non vengono trovati file.

L'esempio seguente esegue il comando COPY con manifest nell'esempio precedente, che viene denominato `cust.manifest`.

```
copy customer
from 's3://mybucket/cust.manifest'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
manifest;
```

Utilizzo di un manifest creato da UNLOAD

Un manifest creato da un'operazione UNLOAD utilizzando il parametro MANIFEST potrebbe avere chiavi non richieste per l'operazione di COPY. Ad esempio, il manifest `UNLOAD` seguente include una chiave `meta` necessaria per una tabella esterna di Amazon Redshift Spectrum e per il caricamento di file di dati in formato `ORC` o `Parquet`. La chiave `meta` contiene una chiave `content_length` con un valore che rappresenta le dimensioni effettive del file in byte. L'operazione di COPY richiede solo la chiave `url` e una chiave opzionale `mandatory`.

```
{
  "entries": [
```

```
    {"url":"s3://mybucket/unload/manifest_0000_part_00", "meta": { "content_length":
 5956875 }},
    {"url":"s3://mybucket/unload/unload/manifest_0001_part_00", "meta":
 { "content_length": 5997091 }}
 ]
}
```

Per ulteriori informazioni sui file manifest, consulta Example: COPY from Amazon S3 using a manifest.

Caricamento di file di dati compressi da Amazon S3

Per caricare file di dati compressi con gzip, lzop o bzip2, includi l'opzione corrispondente: GZIP, LZOP o BZIP2.

Ad esempio, il seguente comando carica da file compressi utilizzando lzop.

```
copy customer from 's3://mybucket/customer.lzo'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|' lzop;
```

> **Note**
>
> Se comprimi un file di dati con la compressione lzop e usi l'opzione --filter, il comando COPY non la supporta.

Caricamento di dati a larghezza fissa da Amazon S3

I file di dati a larghezza fissa hanno lunghezze uniformi per ciascuna colonna di dati. Ogni campo in un file di dati a larghezza fissa ha esattamente la stessa lunghezza e la stessa posizione. Per i dati di carattere (CHAR e VARCHAR) in un file di dati a larghezza fissa, è necessario includere spazi iniziali o finali come segnaposti per mantenere uniforme la larghezza. Per gli interi, è necessario utilizzare gli zero iniziali come segnaposti. Un file di dati a larghezza fissa non ha delimitatori per separare le colonne.

Per caricare un file di dati a larghezza fissa in una tabella esistente, UTILIZZA il parametro FIXEDWIDTH nel comando COPY. Le specifiche della tabella devono corrispondere al valore di fixedwidth_spec affinché i dati vengano caricati correttamente.

Per caricare dati a larghezza fissa da un file a una tabella, emetti il seguente comando:

```
copy table_name from 's3://mybucket/prefix'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
fixedwidth 'fixedwidth_spec';
```

Il parametro fixedwidth_spec è una stringa che contiene un identificatore per ogni colonna e la larghezza di ogni colonna, separati da due punti. Le coppie **`column:width`** sono delimitate da virgole. L'identificatore può essere qualsiasi elemento desiderato: numeri, lettere o una combinazione dei due. L'identificatore non ha alcuna relazione con la tabella stessa, quindi la specifica deve contenere le colonne nello stesso ordine della tabella.

I seguenti due esempi mostrano la stessa specifica, il primo con identificatori numerici e il secondo con identificatori di stringa:

```
'0:3,1:25,2:12,3:2,4:6'
```

```
'venueid:3,venuename:25,venuecity:12,venuestate:2,venueseats:6'
```

Il seguente esempio mostra dati di esempio a larghezza fissa che potrebbero essere caricati nella tabella VENUE utilizzando le specifiche precedenti:

```
1  Toyota Park               Bridgeview  IL0
2  Columbus Crew Stadium     Columbus    OH0
3  RFK Stadium               Washington  DC0
4  CommunityAmerica Ballpark Kansas City KS0
5  Gillette Stadium          Foxborough  MA68756
```

Il seguente comando COPY carica questo set di dati nella tabella VENUE:

```
copy venue
from 's3://mybucket/data/venue_fw.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
fixedwidth 'venueid:3,venuename:25,venuecity:12,venuestate:2,venueseats:6';
```

Caricamento di dati multibyte da Amazon S3

Se i dati includono caratteri multibyte non ASCII (ad esempio caratteri cinesi o cirillici), è necessario caricare i dati nelle colonne VARCHAR. Il tipo di dati VARCHAR supporta caratteri UTF-8 a quattro byte, ma il tipo di dati CHAR accetta solo caratteri ASCII a byte singolo. Non è possibile caricare

caratteri a cinque byte o più lunghi nelle tabelle Amazon Redshift. Per ulteriori informazioni su CHAR e VARCHAR, consulta Tipi di dati.

Per verificare quale codifica utilizza un file di input, utilizza il comando *file* di Linux:

```
$ file ordersdata.txt
ordersdata.txt: ASCII English text
$ file uni_ordersdata.dat
uni_ordersdata.dat: UTF-8 Unicode text
```

Caricamento di file di dati crittografati da Amazon S3

È possibile utilizzare il comando COPY per caricare i file di dati che sono stati caricati in Amazon S3 utilizzando la crittografia lato server, la crittografia lato client o entrambe.

Il comando COPY supporta i seguenti tipi di crittografia Amazon S3:

- Crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)
- Crittografia lato server con (SSE-KMS) AWS KMS keys
- Crittografia lato server con una chiave root simmetrica lato client

Il comando COPY non supporta i seguenti tipi di crittografia Amazon S3:

- Crittografia lato server con chiavi fornite dal cliente (SSE-C)
- Crittografia lato client utilizzando un AWS KMS key
- Crittografia lato client utilizzando una chiave root asimmetrica fornita dal cliente

Per ulteriori informazioni sulla crittografia Amazon S3, consultare Protezione dei dati con la crittografia lato server e Protezione dei dati con la crittografia lato client nella Guida per l'utente di Amazon Simple Storage Service.

Il comando UNLOAD crittografa automaticamente i file utilizzando SSE-S3. È inoltre possibile scaricare utilizzando la crittografia SSE-KMS o la crittografia lato client con una chiave simmetrica gestita dal cliente. Per ulteriori informazioni, consulta Scaricamento di file di dati crittografati

Il comando COPY riconosce e carica automaticamente i file crittografati utilizzando SSE-S3 e SSE-KMS. È possibile caricare i file crittografati utilizzando una chiave root simmetrica lato client specificando l'opzione ENCRYPTED e fornendo il valore della chiave. Per ulteriori informazioni, consulta Caricamento di dati crittografati su Amazon S3.

Per caricare i file di dati crittografati lato client, fornisci il valore della chiave root utilizzando il parametro MASTER_SYMMETRIC_KEY e includi l'opzione ENCRYPTED.

```
copy customer from 's3://mybucket/encrypted/customer'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
master_symmetric_key '<root_key>'
encrypted
delimiter '|';
```

Per caricare file di dati crittografati compressi gzip, lzop o bzip2, includi l'opzione GZIP, LZOP o BZIP2 insieme al valore della chiave root e all'opzione ENCRYPTED.

```
copy customer from 's3://mybucket/encrypted/customer'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
master_symmetric_key '<root_key>'
encrypted
delimiter '|'
gzip;
```

Caricamento di dati da Amazon EMR

È possibile utilizzare il comando COPY per caricare dati in parallelo da un cluster Amazon EMR configurato per scrivere file di testo nel File system distribuito Hadoop (HDFS) del cluster sotto forma di file a larghezza fissa, delimitati da caratteri, CSV o formattati JSON.

Processo per il caricamento dei dati da Amazon EMR

In questa sezione viene illustrato il processo di caricamento dei dati da un cluster Amazon EMR. Le sezioni seguenti forniscono le informazioni dettagliate necessarie per completare ogni fase.

- Fase 1: configurazione delle autorizzazioni IAM

  Gli utenti che creano il cluster Amazon EMR ed eseguono il comando COPY di Amazon Redshift devono disporre delle autorizzazioni necessarie.
- Fase 2: Creazione di un cluster Amazon EMR

  Configurare il cluster per i file di testo di output su Hadoop Distributed File System (HDFS). Saranno necessari l'ID del cluster Amazon EMR e il DNS pubblico principale del cluster (l'endpoint per l'istanza Amazon EC2 che ospita il cluster).

- Fase 3: Recupero degli indirizzi IP dei nodi del cluster e della chiave pubblica del cluster Amazon Redshift

  La chiave pubblica consente ai nodi del cluster Amazon Redshift per stabilire connessioni SSH agli host. Sarà utilizzato l'indirizzo IP per ciascun nodo del cluster per configurare i gruppi di sicurezza dell'host in modo da consentire l'accesso dal cluster Amazon Redshift utilizzando questi indirizzi IP.
- Fase 4: Aggiunta della chiave pubblica del cluster Amazon Redshift a ciascun file di chiavi autorizzate dell'host Amazon EC2

  Aggiungere la chiave pubblica del cluster Amazon Redshift al file di chiavi autorizzate dell'host in modo che l'host riconosca il cluster Amazon Redshift e accetti la connessione SSH.
- Fase 5: Configurazione degli host affinché accettino tutti gli indirizzi IP del cluster Amazon Redshift

  Modifica i gruppi di sicurezza dell'istanza Amazon EMR per aggiungere regole di input per accettare gli indirizzi IP di Amazon Redshift.
- Fase 6. esecuzione del comando COPY per il caricamento di dati

  Da un database Amazon Redshift, eseguire il comando COPY per caricare i dati in una tabella Amazon Redshift.

Fase 1: configurazione delle autorizzazioni IAM

Gli utenti che creano il cluster Amazon EMR ed eseguono il comando COPY di Amazon Redshift devono disporre delle autorizzazioni necessarie.

Per configurare le autorizzazioni IAM

1. Aggiungi le seguenti autorizzazioni per l'utente che creerà il cluster Amazon EMR.

   ```
   ec2:DescribeSecurityGroups
   ec2:RevokeSecurityGroupIngress
   ec2:AuthorizeSecurityGroupIngress
   redshift:DescribeClusters
   ```

2. Aggiungi le seguenti autorizzazioni per l'utente o il ruolo IAM che eseguirà il comando COPY.

   ```
   elasticmapreduce:ListInstances
   ```

3. Aggiungere le seguenti autorizzazioni al ruolo IAM del cluster Amazon EMR.

```
redshift:DescribeClusters
```

Fase 2: Creazione di un cluster Amazon EMR

Il comando COPY carica i dati dai file nell'file di sistema distribuito Hadoop (HDFS) di Amazon EMR. Quando si crei il cluster Amazon EMR, configurare il cluster per i file di dati di output nell'HDFS del cluster.

Come creare un cluster Amazon EMR

1. Crea un cluster Amazon EMR nella stessa AWS regione del cluster Amazon Redshift.

   Se il cluster Amazon Redshift si trova in un VPC, il cluster Amazon EMR dovrà trovarsi nello stesso gruppo VPC. Se il cluster Amazon Redshift utilizza la modalità EC2-Classic (cioè, non si trova in un VPC), anche il cluster Amazon EMR dovrà utilizzare la modalità EC2-Classic. Per ulteriori informazioni, consulta Gestione dei cluster nel cloud privato virtuale (VPC) nella Guida alla gestione di Amazon Redshift.
2. Configurare il cluster per i file di dati di output nell'HDFS del cluster. I nomi dei file HDFS non devono contenere asterischi (*) o punti interrogativi (?).

   > **Important**
   >
   > I nomi dei file non devono contenere asterischi (*) o punti interrogativi (?).

3. Specificare No per l'opzione Termina automaticamente nella configurazione del cluster Amazon EMR in modo che il cluster rimanga disponibile mentre viene eseguito il comando COPY.

   > **Important**
   >
   > Se uno qualsiasi dei file di dati viene modificato o cancellato prima del completamento dell'operazione di COPY, si potrebbero ottenere risultati imprevisti o l'operazione di COPY potrebbe fallire.

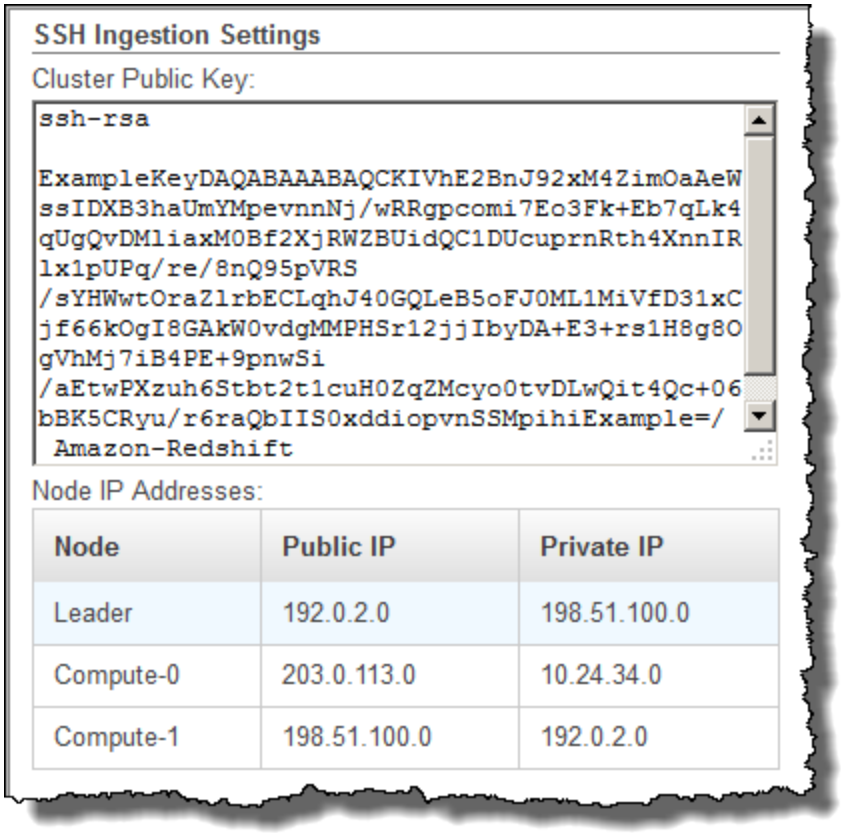
4. Prendi nota dell'ID e del DNS pubblico principale del cluster (l'endpoint per l'istanza Amazon EC2 che ospita il cluster). Queste informazioni saranno utili per le fasi successive.

Fase 3: Recupero degli indirizzi IP dei nodi del cluster e della chiave pubblica del cluster Amazon Redshift

Come recuperare la chiave pubblica del cluster Amazon Redshift e gli indirizzi IP dei nodi del cluster tramite la console

1. Accedere alla console di gestione di Amazon Redshift.
2. Nel riquadro di navigazione scegli Clusters (Cluster).
3. Selezionare il cluster dall'elenco.
4. Individuare il gruppo SSH Ingestion Settings (Impostazioni di inserimento SSH).

   Prendere nota dei valori di Cluster Public Key (Chiave pubblica del cluster) e Node IP addresses (Indirizzi IP del nodo). Queste informazioni saranno utili per le fasi successive.



   Saranno utilizzati gli indirizzi IP privati nella fase 3 per configurare l'host Amazon EC2 affinché accetti la connessione da Amazon Redshift.

Per recuperare la chiave pubblica del cluster e gli indirizzi IP dei nodi del cluster tramite la CLI di Amazon Redshift, emettere il comando describe-clusters. Per esempio:

```
aws redshift describe-clusters --cluster-identifier <cluster-identifier>
```

La risposta includerà un ClusterPublicKey valore e l'elenco di indirizzi IP privati e pubblici, simili al seguente:

```
{
    "Clusters": [
        {
            "VpcSecurityGroups": [],
            "ClusterStatus": "available",
            "ClusterNodes": [
                {
                    "PrivateIPAddress": "10.nnn.nnn.nnn",
                    "NodeRole": "LEADER",
                    "PublicIPAddress": "10.nnn.nnn.nnn"
                },
                {
                    "PrivateIPAddress": "10.nnn.nnn.nnn",
                    "NodeRole": "COMPUTE-0",
                    "PublicIPAddress": "10.nnn.nnn.nnn"
                },
                {
                    "PrivateIPAddress": "10.nnn.nnn.nnn",
                    "NodeRole": "COMPUTE-1",
                    "PublicIPAddress": "10.nnn.nnn.nnn"
                }
            ],
            "AutomatedSnapshotRetentionPeriod": 1,
            "PreferredMaintenanceWindow": "wed:05:30-wed:06:00",
            "AvailabilityZone": "us-east-1a",
            "NodeType": "dc2.large",
            "ClusterPublicKey": "ssh-rsa AAAABexamplepublickey...Y3TAl Amazon-
Redshift",
             ...
             ...
}
```

Per recuperare la chiave pubblica del cluster e gli indirizzi IP dei nodi del cluster per il cluster con l'API di Amazon Redshift, utilizzare l'operazione `DescribeClusters`. Per ulteriori informazioni,

consulta describe-clusters nella Amazon Redshift CLI Guide o DescribeClustersnella Amazon Redshift API Guide.

Fase 4: Aggiunta della chiave pubblica del cluster Amazon Redshift a ciascun file di chiavi autorizzate dell'host Amazon EC2

Aggiungere la chiave pubblica del cluster a ciascun file delle chiavi autorizzate dell'host per tutti i nodi del cluster Amazon EMR in modo che gli host riconoscano Amazon Redshift e accettino la connessione SSH.

Come aggiungere la chiave pubblica del cluster Amazon Redshift al file di chiavi autorizzate di ciascun host

1. Accedere all'host tramite una connessione SSH.

   Per ulteriori informazioni sulla connessione a un'istanza che utilizza SSH, consultare Connessione all'istanza nella Guida per l'utente di Amazon EC2.
2. Copiare la chiave pubblica di Amazon Redshift dalla console o dal testo di risposta della CLI.
3. Copiare e incollare i contenuti della chiave pubblica nel file `/home/<ssh_username>/.ssh/authorized_keys` nell'host. Includere la stringa completa, compreso il prefisso "`ssh-rsa` " e il suffisso "`Amazon-Redshift`". Ad esempio:

   ```
   ssh-rsa AAAACTP3isxgGzVWoIWpbVvRCOzYdVifMrh… uA70BnMHCaMiRdmvsDOedZDOedZ Amazon-
   Redshift
   ```

Fase 5: Configurazione degli host affinché accettino tutti gli indirizzi IP del cluster Amazon Redshift

Per consentire il traffico in entrata verso le istanze dell'host, modificare il gruppo di sicurezza e aggiungere una regola in entrata per ciascun nodo del cluster Amazon Redshift. In Type (Tipo), seleziona SSH con il protocollo TCP sulla porta 22. In Source (Origine), inserisci gli indirizzi IP privati dei nodi cluster di Amazon Redshift recuperati in Fase 3: Recupero degli indirizzi IP dei nodi del cluster e della chiave pubblica del cluster Amazon Redshift. Per informazioni sull'aggiunta di regole a un gruppo di sicurezza Amazon EC2, consultare Autorizzazione del traffico in entrata per le istanze nella Guida per l'utente di Amazon EC2.

Fase 6. esecuzione del comando COPY per il caricamento di dati

Eseguire un comando COPY per effettuare la connessione al cluster Amazon EMR e caricare i dati in una tabella Amazon Redshift. Il cluster Amazon EMR deve continuare a funzionare fino al completamento del comando COPY. Ad esempio, non configurare la terminazione automatica del cluster.

> **Important**
>
> Se uno qualsiasi dei file di dati viene modificato o cancellato prima del completamento dell'operazione di COPY, si potrebbero ottenere risultati imprevisti o l'operazione di COPY potrebbe fallire.

Nel comando COPY, specificare l'ID cluster Amazon EMR e il percorso del file HDFS e il nome del file.

```
copy sales
from 'emr://myemrclusterid/myoutput/part*' credentials
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

È possibile utilizzare i caratteri jolly asterisco (*) e punto interrogativo (?) come parte dell'argomento del nome file. Ad esempio, `part*` carica i file `part-0000`, `part-0001` e così via. Se specifichi solo il nome di una cartella, COPY tenta di caricare tutti i file nella cartella.

> **Important**
>
> Se utilizzi caratteri jolly o solo il nome della cartella, verifica che non vengano caricati file indesiderati o che il comando COPY fallisca. Ad esempio, alcuni processi potrebbero scrivere un file di log nella cartella di output.

Caricamento di dati da host remoti

È possibile utilizzare il comando COPY per caricare dati in parallelo da uno o più host remoti, quali istanze Amazon EC2 o altri computer. COPY si connette agli host remoti utilizzando SSH ed esegue comandi sugli host remoti per generare output di testo.

L'host remoto può essere un'istanza Linux di Amazon EC2 o un altro computer Unix o Linux configurato per accettare connessioni SSH. Questa guida presuppone che l'host remoto si trovi in un'istanza Amazon EC2. Se la procedura è diversa per un altro computer, la guida indicherà la differenza.

Amazon Redshift può connettersi a più host e può aprire più connessioni SSH per ogni host. Amazon Redshift invia un comando univoco attraverso ogni connessione per generare output di testo per l'output standard dell'host, che legge quindi come un file di testo.

Prima di iniziare

Prima di iniziare, devi disporre dei seguenti requisiti:

- Una o più macchine host, come le istanze Amazon EC2, a cui è possibile effettuare la connessione utilizzando SSH.
- Origini dati negli host.

  Saranno forniti i comandi che il cluster Amazon Redshift eseguirà sugli host per generare l'output di testo. Una volta che il cluster si connette a un host, il comando COPY esegue i comandi, legge il testo dall'output standard degli host e carica i dati in parallelo in una tabella Amazon Redshift. L'output di testo deve essere in un formato importabile dal comando COPY. Per ulteriori informazioni, consulta Preparazione dei dati di input
- Accesso agli host dal computer.

  Per un'istanza Amazon EC2, per accedere all'host verrà utilizzata una connessione SSH. Devi accedere all'host per aggiungere la chiave pubblica del cluster Amazon Redshift al file di chiavi autorizzate dell'host.
- Un cluster Amazon Redshift in esecuzione.

  Per informazioni su come avviare un cluster, consultare Guida alle operazioni di base di Amazon Redshift.

Processo di caricamento dei dati

Questa sezione illustra il processo di caricamento dei dati da host remoti. Le sezioni seguenti forniscono le informazioni dettagliate necessarie per completare ogni fase.

- Fase 1: recupero degli indirizzi IP dei nodi del cluster e della chiave pubblica del cluster

La chiave pubblica consente ai nodi del cluster Amazon Redshift di stabilire connessioni SSH agli host remoti. Sarà utilizzato l'indirizzo IP per ciascun nodo del cluster per configurare i gruppi di sicurezza dell'host o del firewall in modo da consentire l'accesso dal cluster Amazon Redshift utilizzando questi indirizzi IP.

- Fase 2: Aggiunta della chiave pubblica del cluster Amazon Redshift al file di chiavi autorizzate di ciascun host

  Aggiungere la chiave pubblica del cluster Amazon Redshift al file di chiavi autorizzate dell'host in modo che l'host riconosca il cluster Amazon Redshift e accetti la connessione SSH.

- Fase 3: Configurazione dell'host affinché accetti tutti gli indirizzi IP del cluster Amazon Redshift

  Per Amazon EC2, modifica i gruppi di sicurezza dell'istanza per aggiungere regole di input per accettare gli indirizzi IP di Amazon Redshift. Per gli altri host, modificare il firewall in modo che i nodi Amazon Redshift possano stabilire connessioni SSH all'host remoto.

- Fase 4: ottenere una chiave pubblica per l'host

  È possibile specificare che Amazon Redshift deve utilizzare la chiave pubblica per identificare l'host. Devi individuare la chiave pubblica e copiare il testo nel file manifesto.

- Fase 5: creazione di un file manifest

  Il manifest è un file di testo in formato JSON con i dettagli necessari ad Amazon Redshift per effettuare la connessione agli host e recuperare i dati.

- Fase 6: Caricamento del file manifest in un bucket Amazon S3

  Amazon Redshift legge il manifest e utilizza tali informazioni per effettuare la connessione all'host remoto. Se il bucket Amazon S3 non si trova nella stessa regione del cluster Amazon Redshift, è necessario utilizzare l'opzione REGION per specificare la regione in cui si trovano i dati.

- Fase 7: esecuzione del comando COPY per il caricamento di dati

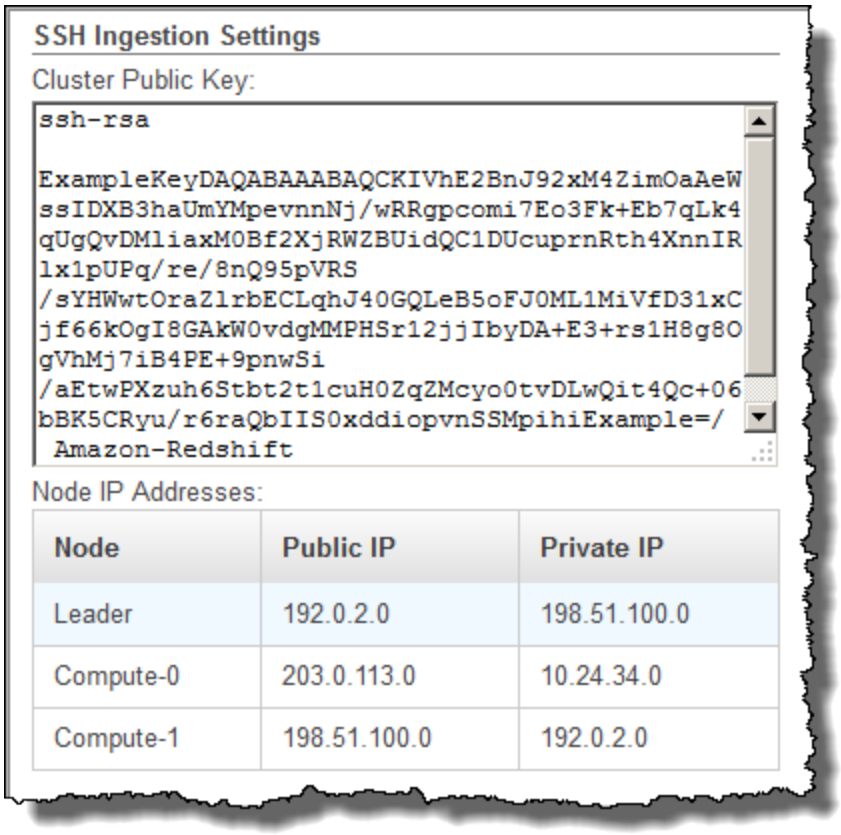
  Da un database Amazon Redshift, eseguire il comando COPY per caricare i dati in una tabella Amazon Redshift.

Fase 1: recupero degli indirizzi IP dei nodi del cluster e della chiave pubblica del cluster

Per recuperare la chiave pubblica del cluster e gli indirizzi IP dei nodi del cluster tramite la console

1. Accedere alla console di gestione di Amazon Redshift.
2. Nel riquadro di navigazione scegli Clusters (Cluster).
3. Selezionare il cluster dall'elenco.
4. Individuare il gruppo SSH Ingestion Settings (Impostazioni di inserimento SSH).

   Prendere nota dei valori di Cluster Public Key (Chiave pubblica del cluster) e Node IP addresses (Indirizzi IP del nodo). Queste informazioni saranno utili per le fasi successive.



   Per configurare l'host affinché accetti la connessione da Amazon Redshift, saranno utilizzati gli indirizzi IP della fase 3. A seconda del tipo di host a cui ti connetti e se si trova in un VPC, utilizzerai gli indirizzi IP pubblici o gli indirizzi IP privati.

Per recuperare la chiave pubblica del cluster e gli indirizzi IP dei nodi del cluster tramite la CLI di Amazon Redshift, emettere il comando describe-clusters.

Per esempio:

```
aws redshift describe-clusters --cluster-identifier <cluster-identifier>
```

La risposta includerà l'elenco ClusterPublicKey e l'elenco di indirizzi IP privati e pubblici, simili ai seguenti:

```
{
    "Clusters": [
        {
            "VpcSecurityGroups": [],
            "ClusterStatus": "available",
            "ClusterNodes": [
                {
                    "PrivateIPAddress": "10.nnn.nnn.nnn",
                    "NodeRole": "LEADER",
                    "PublicIPAddress": "10.nnn.nnn.nnn"
                },
                {
                    "PrivateIPAddress": "10.nnn.nnn.nnn",
                    "NodeRole": "COMPUTE-0",
                    "PublicIPAddress": "10.nnn.nnn.nnn"
                },
                {
                    "PrivateIPAddress": "10.nnn.nnn.nnn",
                    "NodeRole": "COMPUTE-1",
                    "PublicIPAddress": "10.nnn.nnn.nnn"
                }
            ],
            "AutomatedSnapshotRetentionPeriod": 1,
            "PreferredMaintenanceWindow": "wed:05:30-wed:06:00",
            "AvailabilityZone": "us-east-1a",
            "NodeType": "dc2.large",
            "ClusterPublicKey": "ssh-rsa AAAABexamplepublickey...Y3TAl Amazon-
Redshift",
             ...
             ...
}
```

Per recuperare la chiave pubblica del cluster e gli indirizzi IP del nodo del cluster per il cluster utilizzando l'API Amazon Redshift, utilizza l' DescribeClusters azione. Per ulteriori informazioni, consulta describe-clusters nella Amazon Redshift CLI Guide o DescribeClustersnella Amazon Redshift API Guide.

Fase 2: Aggiunta della chiave pubblica del cluster Amazon Redshift al file di chiavi autorizzate di ciascun host

Aggiungere la chiave pubblica del cluster a ciascun file delle chiavi autorizzate dell'host in modo che l'host riconosca il cluster Amazon Redshift e accetti la connessione SSH.

Come aggiungere la chiave pubblica del cluster Amazon Redshift al file di chiavi autorizzate di ciascun host

1. Accedere all'host tramite una connessione SSH.

   Per ulteriori informazioni sulla connessione a un'istanza che utilizza SSH, consultare Connessione all'istanza nella Guida per l'utente di Amazon EC2.
2. Copiare la chiave pubblica di Amazon Redshift dalla console o dal testo di risposta della CLI.
3. Copiare e incollare i contenuti della chiave pubblica nel file `/home/<ssh_username>/.ssh/authorized_keys` nell'host remoto. Il valore `<ssh_username>` deve corrispondere al valore per il campo "username" nel file manifest. Includere la stringa completa, compreso il prefisso "`ssh-rsa` " e il suffisso "`Amazon-Redshift`". Ad esempio:

   ```
   ssh-rsa AAAACTP3isxgGzVWoIWpbVvRCOzYdVifMrh… uA70BnMHCaMiRdmvsDOedZDOedZ Amazon-
   Redshift
   ```

Fase 3: Configurazione dell'host affinché accetti tutti gli indirizzi IP del cluster Amazon Redshift

Se si lavora con un'istanza Amazon EC2 o un cluster Amazon EMR, aggiungere le regole in entrata al gruppo di sicurezza dell'host per consentire il traffico da ogni nodo del cluster Amazon Redshift. In Type (Tipo), seleziona SSH con il protocollo TCP sulla porta 22. In Origine, inserire gli indirizzi IP dei nodi del cluster di Amazon Redshift recuperati in Fase 1: recupero degli indirizzi IP dei nodi del cluster e della chiave pubblica del cluster. Per informazioni sull'aggiunta di regole a un gruppo di sicurezza Amazon EC2, consultare Autorizzazione del traffico in entrata per le istanze nella Guida per l'utente di Amazon EC2.

Utilizza gli indirizzi IP privati quando:

- Hai un cluster Amazon Redshift che non si trova in un Virtual Private Cloud (VPC) e un'istanza Amazon EC2 -Classic, entrambi situati nella stessa regione. AWS
- Hai un cluster Amazon Redshift che si trova in un VPC e un'istanza Amazon EC2 -VPC, entrambi nella stessa regione e nello stesso VPC. AWS

In alternativa, utilizza gli indirizzi IP pubblici.

Per ulteriori informazioni sull'utilizzo di Amazon Redshift in un VPC, consulta Gestione dei cluster nel cloud privato virtuale (VPC) nella Guida alla gestione di Amazon Redshift.

Fase 4: ottenere una chiave pubblica per l'host

Facoltativamente, è possibile fornire la chiave pubblica dell'host nel file manifest in modo che Amazon Redshift possa identificare l'host. Il comando COPY non richiede la chiave pubblica dell'host ma, per motivi di sicurezza, consigliamo di utilizzare una chiave pubblica per prevenire attacchi "man-in-the-middle".

È possibile trovare la chiave pubblica dell'host nel seguente percorso, dove `<ssh_host_rsa_key_name>` è il nome univoco della chiave pubblica dell'host:

```
:  /etc/ssh/<ssh_host_rsa_key_name>.pub
```

> **Note**
>
> Amazon Redshift supporta solo le chiavi RSA. Non supportiamo le chiavi DSA.

Quando crei il file manifest alla fase 5, incollerai il testo della chiave pubblica nel campo "Public Key" (Chiave pubblica) nella voce del file manifest.

Fase 5: creazione di un file manifest

Il comando COPY può connettersi a più host tramite SSH e può creare più connessioni SSH per ogni host. COPY esegue un comando attraverso ogni connessione host, quindi carica l'output dai comandi in parallelo nella tabella. Il file manifest è un file di testo in formato JSON che Amazon Redshift utilizza per la connessione all'host. Il file manifest specifica gli endpoint dell'host SSH e i comandi che

vengono eseguiti sugli host per restituire i dati ad Amazon Redshift. Facoltativamente, puoi includere la chiave pubblica dell'host, il nome utente di login e un flag obbligatorio per ogni voce.

Crea il file manifest nel computer locale. In un passaggio successivo, il file viene caricato in Amazon S3.

Il file manifest è nel seguente formato:

```
{
   "entries": [
     {"endpoint":"<ssh_endpoint_or_IP>",
       "command": "<remote_command>",
       "mandatory":true,
       "publickey": "<public_key>",
       "username": "<host_user_name>"},
     {"endpoint":"<ssh_endpoint_or_IP>",
       "command": "<remote_command>",
       "mandatory":true,
       "publickey": "<public_key>",
       "username": "host_user_name"}
    ]
}
```

Il file manifest contiene un costrutto "entries" per ogni connessione SSH. Ogni voce rappresenta una singola connessione SSH. È possibile avere più connessioni a un singolo host o più connessioni a più host. Le doppie virgolette sono richieste come mostrato, sia per i nomi dei campi sia per i valori. L'unico valore che non necessita di doppie virgolette è il valore booleano **`true`** o **`false`** per il campo obbligatorio.

Di seguito sono descritti i campi del file manifest.

endpoint

L'indirizzo URL o l'indirizzo IP dell'host. Ad esempio, "`ec2-111-222-333.compute-1.amazonaws.com`" o "`22.33.44.56`"

command

Il comando che verrà eseguito dall'host per generare output di testo o binario (gzip, lzop, or bzip2). Il comando può essere qualsiasi comando che l'utente "host_user_name" è autorizzato a eseguire. Il comando può essere semplice come stampare un file o eseguire una query su

un database o lanciare uno script. L'output (file di testo o file binario gzip, lzop o bzip2) deve essere in un formato che il comando COPY di Amazon Redshift possa importare. Per ulteriori informazioni, consulta Preparazione dei dati di input.

publickey

(Facoltativo) La chiave pubblica dell'host. Se fornita, Amazon Redshift userà la chiave pubblica per identificare l'host. Se la chiave pubblica non viene fornita, Amazon Redshift non tenterà l'identificazione dell'host. Ad esempio, se la chiave pubblica dell'host remoto è `ssh-rsa AbcCbaxxx…xxxDHKJ root@amazon.com`, digita il seguente testo nel campo della chiave pubblica: `AbcCbaxxx…xxxDHKJ`.

mandatory

(Facoltativo) Indica se il comando COPY debba fallire se la connessione fallisce. Il valore predefinito è `false`. Se Amazon Redshift non riesce a effettuare almeno una connessione, il comando COPY avrà esito negativo.

username

(Facoltativo) Il nome utente che verrà utilizzato per accedere al sistema host ed eseguire il comando remoto. Il nome di login dell'utente deve essere lo stesso utilizzato per aggiungere la chiave pubblica al file delle chiavi autorizzate nella fase 2. Il nome utente predefinito è "redshift".

Il seguente esempio mostra un manifest completato per aprire quattro connessioni allo stesso host ed eseguire un comando diverso tramite ciascuna connessione:

```
{
  "entries": [
       {"endpoint":"ec2-184-72-204-112.compute-1.amazonaws.com",
          "command": "cat loaddata1.txt",
          "mandatory":true,
          "publickey": "ec2publickeyportionoftheec2keypair",
          "username": "ec2-user"},
       {"endpoint":"ec2-184-72-204-112.compute-1.amazonaws.com",
          "command": "cat loaddata2.txt",
          "mandatory":true,
          "publickey": "ec2publickeyportionoftheec2keypair",
          "username": "ec2-user"},
       {"endpoint":"ec2-184-72-204-112.compute-1.amazonaws.com",
          "command": "cat loaddata3.txt",
          "mandatory":true,
```

```
          "publickey": "ec2publickeyportionoftheec2keypair",
          "username": "ec2-user"},
       {"endpoint":"ec2-184-72-204-112.compute-1.amazonaws.com",
          "command": "cat loaddata4.txt",
          "mandatory":true,
          "publickey": "ec2publickeyportionoftheec2keypair",
          "username": "ec2-user"}
     ]
}
```

Fase 6: Caricamento del file manifest in un bucket Amazon S3

Caricare il file manifest in un bucket Amazon S3. Se il bucket Amazon S3 non si trova nella stessa AWS regione del cluster Amazon Redshift, devi utilizzare l'REGIONopzione per specificare la AWS regione in cui si trova il manifest. Per ulteriori informazioni sulla creazione di un bucket Amazon S3 e sul caricamento di un file, consultare Guida per l'utente di Amazon Simple Storage Service.

Fase 7: esecuzione del comando COPY per il caricamento di dati

Eseguire un comando COPY per effettuare la connessione all'host e caricare i dati in una tabella Amazon Redshift. Nel comando COPY, specificare il percorso dell'oggetto Amazon S3 esplicito per il file manifest e includere l'opzione SSH. Ad esempio,

```
copy sales
from 's3://mybucket/ssh_manifest'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|'
ssh;
```

> **Note**
>
> Se utilizzi la compressione automatica, il comando COPY esegue due operazioni di lettura dati, il che significa che eseguirà il comando remoto due volte. La prima operazione di lettura consiste nel fornire un campione per l'analisi della compressione e la seconda operazione di lettura carica effettivamente i dati. Se la doppia esecuzione del comando remoto può causare un problema, dati i potenziali effetti collaterali, è necessario disattivare la compressione automatica. Per disattivare la compressione automatica, esegui il comando COPY con l'opzione COMPUPDATE impostata su OFF. Per ulteriori informazioni, consulta Caricamento di tabelle con compressione automatica.

Caricamento di dati da una tabella Amazon DynamoDB

È possibile utilizzare il comando COPY per caricare una tabella con dati da una singola tabella Amazon DynamoDB.

> **Important**
>
> La tabella Amazon DynamoDB che fornisce i dati deve essere creata nella AWS stessa regione del cluster a meno che non si utilizzi l'opzione per specificare REGION la regione in cui si trova AWS la tabella Amazon DynamoDB.

Il comando COPY utilizza l'architettura MPP (Massively Parallel Processing) di Amazon Redshift per leggere e caricare dati in parallelo da una tabella Amazon DynamoDB. È possibile ottenere il massimo dall'elaborazione parallela impostando gli stili di distribuzione nelle tabelle Amazon Redshift. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati.

> **Important**
>
> Quando il comando COPY legge i dati dalla tabella Amazon DynamoDB, il trasferimento dei dati risultante fa parte del throughput assegnato di quella tabella.

Per evitare di utilizzare quantità eccessive di throughput di lettura assegnato, consigliamo di non caricare i dati dalle tabelle Amazon DynamoDB che si trovano in ambienti di produzione. Se carichi i dati dalle tabelle di produzione, consigliamo di impostare l'opzione READRATIO su un valore molto più basso della percentuale media del throughput assegnato inutilizzato. Un'impostazione READRATIO bassa contribuirà a ridurre al minimo i problemi di throttling. Per utilizzare l'intero throughput assegnato di una tabella Amazon DynamoDB, impostare READRATIO su 100.

Il comando COPY fa corrispondere i nomi degli attributi negli elementi recuperati dalla tabella DynamoDB ai nomi di colonna nella tabella Amazon Redshift esistente utilizzando le seguenti regole:

- Le colonne di una tabella Amazon Redshift corrispondono senza distinzione tra maiuscole e minuscole agli attributi degli elementi di Amazon DynamoDB. Se una voce nella tabella DynamoDB contiene più attributi che differiscono solo tra maiuscole e minuscole, come Price e PRICE, il comando COPY fallirà.

- Le colonne della tabella Amazon Redshift che non corrispondono a un attributo nella tabella Amazon DynamoDB vengono caricate come NULL o vuote, a seconda del valore specificato con l'opzione EMPTYASNULL nel comando COPY.
- Gli attributi di Amazon DynamoDB che non corrispondono a una colonna nella tabella Amazon Redshift vengono scartati. Gli attributi vengono letti prima di essere abbinati, quindi anche gli attributi eliminati utilizzano parte del throughput assegnato della tabella.
- Sono supportati solo gli attributi di Amazon DynamoDB con tipi di dati STRING e NUMBER scalari. I tipi di dati BINARY e SET di Amazon DynamoDB non sono supportati. Se un comando COPY prova a caricare un attributo con un tipo di dati non supportato, il comando fallirà. Se l'attributo non corrisponde a una colonna della tabella Amazon Redshift, il comando COPY non prova a caricarlo e non genera un errore.

Il comando COPY utilizza la seguente sintassi per caricare i dati da una tabella Amazon DynamoDB:

```
copy <redshift_tablename> from 'dynamodb://<dynamodb_table_name>'
authorization
readratio '<integer>';
```

I valori per l'autorizzazione sono le AWS credenziali necessarie per accedere alla tabella Amazon DynamoDB. Se queste credenziali corrispondono a un utente, tale utente deve disporre dell'autorizzazione per le operazioni SCAN e DESCRIBE per la tabella Amazon DynamoDB che viene caricata.

I valori di autorizzazione forniscono l' AWS autorizzazione necessaria al cluster per accedere alla tabella Amazon DynamoDB. L'autorizzazione deve includere SCAN e DESCRIBE per la tabella Amazon DynamoDB che viene caricata. Per ulteriori informazioni sulle autorizzazioni richieste, consultare Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY. Il metodo preferito per l'autenticazione è specificare il parametro IAM_ROLE e fornire l'Amazon Resource Name (ARN) per un ruolo IAM con le autorizzazioni necessarie. Per ulteriori informazioni, consulta Controllo degli accessi basato sui ruoli.

Per effettuare l'autenticazione utilizzando il parametro IAM_ROLE, utilizza *<aws-account-id>* e *<role-name>* come mostrato nella sintassi seguente.

```
IAM_ROLE 'arn:aws:iam::<aws-account-id>:role/<role-name>'
```

L'esempio seguente mostra l'autenticazione utilizzando un ruolo IAM.

```
copy favoritemovies
from 'dynamodb://ProductCatalog'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Per ulteriori informazioni su altre opzioni di autorizzazione, consulta Parametri di autorizzazione

Se desideri convalidare i dati senza caricare effettivamente la tabella, utilizza l'opzione NOLOAD con il comando COPY.

L'esempio seguente carica la tabella FAVORITEMOVIES con i dati della tabella DynamoDB. my-favorite-movies-table L'attività di lettura può utilizzare fino al 50% del throughput assegnato.

```
copy favoritemovies from 'dynamodb://my-favorite-movies-table'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
readratio 50;
```

Per massimizzare il throughput, il comando COPY carica i dati da una tabella Amazon DynamoDB in parallelo in tutti i nodi di calcolo nel cluster.

Throughput assegnata con la compressione automatica

Per impostazione predefinita, il comando COPY applica la compressione automatica ogni volta che specifichi una tabella di destinazione vuota senza codifica di compressione. L'analisi della compressione automatica inizialmente campiona un numero elevato di righe dalla tabella Amazon DynamoDB. La dimensione del campione si basa sul valore del parametro COMPROWS. La versione predefinita è di 100.000 per sezione.

Dopo il campionamento, le righe di esempio vengono scartate e viene caricata l'intera tabella. Di conseguenza, molte righe vengono lette due volte. Per ulteriori informazioni sul funzionamento della compressione automatica, consultare Caricamento di tabelle con compressione automatica.

> **Important**
>
> Quando il comando COPY legge i dati dalla tabella Amazon DynamoDB, incluse le righe utilizzate per il campionamento, il trasferimento dei dati risultante fa parte del throughput assegnato di quella tabella.

Caricamento di dati multibyte da Amazon DynamoDB

Se i dati includono caratteri multibyte non ASCII (ad esempio caratteri cinesi o cirillici), è necessario caricare i dati nelle colonne VARCHAR. Il tipo di dati VARCHAR supporta caratteri UTF-8 a quattro byte, ma il tipo di dati CHAR accetta solo caratteri ASCII a byte singolo. Non è possibile caricare caratteri a cinque byte o più lunghi nelle tabelle Amazon Redshift. Per ulteriori informazioni su CHAR e VARCHAR, consulta Tipi di dati.

Verifica del caricamento corretto dei dati

Al termine dell'operazione di caricamento, eseguire una query sulla tabella del sistema STL_LOAD_COMMITS per verificare che i file previsti siano stati caricati. Esegui il comando COPY e carica la verifica all'interno della stessa transazione in modo che, in caso di problemi con il carico, sia possibile eseguire il rollback dell'intera transazione.

La seguente query restituisce le voci per il caricamento delle tabelle nel database TICKIT:

```
select query, trim(filename) as filename, curtime, status
from stl_load_commits
where filename like '%tickit%' order by query;

 query |         filename          |          curtime           | status
-------+---------------------------+----------------------------+--------
 22475 | tickit/allusers_pipe.txt  | 2013-02-08 20:58:23.274186 |      1
 22478 | tickit/venue_pipe.txt     | 2013-02-08 20:58:25.070604 |      1
 22480 | tickit/category_pipe.txt  | 2013-02-08 20:58:27.333472 |      1
 22482 | tickit/date2008_pipe.txt  | 2013-02-08 20:58:28.608305 |      1
 22485 | tickit/allevents_pipe.txt | 2013-02-08 20:58:29.99489  |      1
 22487 | tickit/listings_pipe.txt  | 2013-02-08 20:58:37.632939 |      1
 22489 | tickit/sales_tab.txt      | 2013-02-08 20:58:37.632939 |      1
(6 rows)
```

Convalida dei dati di input

Per convalidare i dati nei file di input di Amazon S3 o nella tabella Amazon DynamoDB prima di caricare effettivamente i dati, utilizzare l'opzione NOLOAD con il comando COPY. Utilizza NOLOAD con gli stessi comandi e le stesse opzioni di COPY che utilizzeresti per caricare i dati. NOLOAD verifica l'integrità di tutti i dati senza caricarli nel database. L'opzione NOLOAD visualizza eventuali errori che si verificano se avessi provato a caricare i dati.

Ad esempio, se è stato specificato un percorso di Amazon S3 errato per il file di input, Amazon Redshift restituisce il seguente errore.

```
ERROR:  No such file or directory
DETAIL:
-----------------------------------------------
Amazon Redshift error:  The specified key does not exist
code:      2
context:   S3 key being read :
location:  step_scan.cpp:1883
process:   xenmaster [pid=22199]
-----------------------------------------------
```

Per la risoluzione dei messaggi di errore, consultare Riferimento per gli errori di caricamento.

Per un esempio di utilizzo dell'opzione NOLOAD, consulta Comando COPY con l'opzione NOLOAD.

Caricamento di tabelle con compressione automatica

Argomenti

- Come funziona la compressione automatica
- Esempio di compressione automatica

È possibile applicare manualmente le codifiche di compressione alle colonne delle tabelle, in base alla propria valutazione dei dati. In alternativa, è possibile utilizzare il comando COPY con COMPUPDATE impostato su ON per analizzare e applicare automaticamente la compressione in base ai dati di esempio.

È possibile utilizzare la compressione automatica quando crei e carichi una nuova tabella. Il comando COPY esegue un'analisi di compressione. È anche possibile eseguire un'analisi di compressione senza caricare dati o modificare la compressione in una tabella eseguendo il comando ANALYZE COMPRESSION in una tabella già popolata. Ad esempio, è possibile eseguire ANALYZE COMPRESSION quando si desidera analizzare la compressione su una tabella per un utilizzo futuro, mantenendo al contempo le istruzioni DDL (Data Definition Language) esistenti.

La compressione automatica equilibra le prestazioni globali quando si scelgono le codifiche di compressione. Le prestazioni delle scansioni a intervallo limitato potrebbero risultare scadenti se le colonne di chiave di ordinamento vengono compresse più delle altre colonne nella stessa query.

Di conseguenza, la compressione automatica salta la fase di analisi dei dati sulle colonne chiave di ordinamento e mantiene i tipi di codifica definiti dall'utente.

La compressione automatica sceglie la codifica RAW se non è stato definito esplicitamente un tipo di codifica. ANALYZE COMPRESSION si comporta allo stesso modo. Per ottenere prestazioni ottimali delle query, prendere in considerazione l'utilizzo di RAW per le chiavi di ordinamento.

Come funziona la compressione automatica

Quando il parametro COMPUPDATE è impostato su ON, il comando COPY applica la compressione automatica ogni volta che esegui il comando COPY con una tabella di destinazione vuota e tutte le colonne della tabella hanno codifica RAW o nessuna codifica.

Per applicare la compressione automatica a una tabella vuota, indipendentemente dalle codifiche di compressione attuali, esegui il comando COPY con l'opzione COMPUPDATE impostata su ON. Per disattivare la compressione automatica, esegui il comando COPY con l'opzione COMPUPDATE impostata su OFF.

Non è possibile applicare la compressione automatica a una tabella che contiene già dati.

> **Note**
>
> L'analisi della compressione automatica richiede un numero sufficiente di righe nei dati di carico (almeno 100.000 righe per sezione) per generare un campione significativo.

La compressione automatica esegue queste operazioni in background come parte della transazione di caricamento:

1. Un campione iniziale di righe viene caricato dal file di input. La dimensione del campione si basa sul valore del parametro COMPROWS. Il valore predefinito è 100,000.
2. Le opzioni di compressione vengono scelte per ogni colonna.
3. Le righe di esempio vengono rimosse dalla tabella.
4. La tabella viene ricreata con le codifiche di compressione scelte.
5. L'intero file di input viene caricato e compresso utilizzando le nuove codifiche.

Dopo aver eseguito il comando COPY, la tabella viene completamente caricata, compressa ed è pronta per l'uso. Se carichi più dati in un secondo momento, le righe aggiunte vengono compresse in base alla codifica esistente.

Se desideri eseguire solo un'analisi di compressione, esegui ANALYZE COMPRESSION, che è più efficiente rispetto all'esecuzione di un COPY completo. Quindi è possibile valutare i risultati per decidere se utilizzare la compressione automatica o ricreare manualmente la tabella.

La compressione automatica è supportata solo per il comando COPY. In alternativa, è possibile applicare manualmente la codifica della compressione quando crei la tabella. Per informazioni sulla codifica della compressione manuale, consultare Utilizzo della compressione delle colonne.

Esempio di compressione automatica

In questo esempio, presumi che il database TICKIT contenga una copia della tabella LISTING denominata BIGLIST e desideri applicare la compressione automatica a questa tabella quando viene caricata con circa 3 milioni di righe.

Per caricare e comprimere automaticamente la tabella

1. Assicurati che la tabella sia vuota. È possibile applicare la compressione automatica solo a una tabella vuota:

   ```
   truncate biglist;
   ```

2. Caricare la tabella con un singolo comando COPY. Sebbene la tabella sia vuota, potrebbe essere stata specificata una codifica precedente. Per consentire che Amazon Redshift esegua un'analisi di compressione, imposta il parametro COMPUPDATE su ON.

   ```
   copy biglist from 's3://mybucket/biglist.txt'
   iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
   delimiter '|' COMPUPDATE ON;
   ```

   Poiché non viene specificata alcuna opzione COMPROWS, viene utilizzata la dimensione di esempio predefinita e consigliata di 100.000 righe per sezione.

3. Guarda il nuovo schema per la tabella BIGLIST per rivedere gli schemi di codifica scelti automaticamente.

   ```
   select "column", type, encoding
   from pg_table_def where tablename = 'biglist';
   ```

```
     Column     |            Type             | Encoding
----------------+-----------------------------+----------
 listid         | integer                     | az64
 sellerid       | integer                     | az64
 eventid        | integer                     | az64
 dateid         | smallint                    | none
 numtickets     | smallint                    | az64
 priceperticket | numeric(8,2)                | az64
 totalprice     | numeric(8,2)                | az64
 listtime       | timestamp without time zone | az64
```

4. Verifica che il numero previsto di righe sia stato caricato:

```
select count(*) from biglist;

count
---------
3079952
(1 row)
```

Quando le righe vengono aggiunte successivamente a questa tabella utilizzando le istruzioni COPY o INSERT, vengono applicate le stesse codifiche di compressione.

Ottimizzazione dello storage per tabelle limitate

Se disponi di una tabella con poche colonne ma un numero molto elevato di righe, le tre colonne di identità dei metadati nascoste (INSERT_XID, DELETE_XID, ROW_ID) utilizzeranno una quantità sproporzionata di spazio su disco per la tabella.

Per ottimizzare la compressione delle colonne nascoste, carica la tabella in un'unica transazione COPY, dove possibile. Se carichi la tabella con più comandi COPY separati, la colonna INSERT_XID non eseguirà una corretta compressione. Devi eseguire un'operazione vacuum se utilizzi più comandi COPY, ma questa non migliorerà la compressione di INSERT_XID.

Caricamento dei valori delle colonne predefiniti

È possibile definire un elenco di colonne nel comando COPY. Se una colonna della tabella viene omessa dall'elenco di colonne, COPY caricherà la colonna con il valore fornito dall'opzione DEFAULT specificata nel comando CREATE TABLE o con NULL se l'opzione DEFAULT non è stata specificata.

Se COPY tenta di assegnare NULL a una colonna definita come NOT NULL, il comando COPY non viene eseguito. Per informazioni sull'assegnazione dell'opzione DEFAULT, consultare CREATE TABLE.

Quando si esegue il caricamento da file di dati in Amazon S3, le colonne nell'elenco di colonne devono essere nello stesso ordine dei campi nel file di dati. Se un campo nel file di dati non presenta una colonna corrispondente nell'elenco di colonne, il comando COPY fallisce.

Durante il caricamento dalla tabella Amazon DynamoDB l'ordine non è importante. Qualsiasi campo negli attributi di Amazon DynamoDB che non corrisponde a una colonna nella tabella Amazon Redshift sarà eliminato.

Le seguenti restrizioni si applicano quando si utilizza il comando COPY per caricare i valori DEFAULT in una tabella:

- Se una colonna IDENTITY è inclusa nell'elenco delle colonne, l'opzione EXPLICIT_IDS deve essere specificata anche nel comando COPY, altrimenti il comando COPY fallirà. Allo stesso modo, se una colonna IDENTITY viene omessa dall'elenco delle colonne e viene specificata l'opzione EXPLICIT_IDS, l'operazione di COPY fallirà.
- Poiché l'espressione DEFAULT valutata per una determinata colonna è uguale per tutte le righe caricate, un'espressione DEFAULT che utilizza una funzione RANDOM() assegnerà lo stesso valore tutte le righe.
- Le espressioni DEFAULT che contengono CURRENT_DATE o SYSDATE sono impostate sul timestamp della transazione attuale.

Per un esempio, consultare la sezione relativa al caricamento dei dati da un file con valori predefiniti in Esempi di COPY.

Risoluzione di problemi di caricamento dei dati

Argomenti

- ServiceException Errori S3
- Tabelle di sistema per la risoluzione di caricamento dei dati
- Errori di caricamento di caratteri multibyte
- Riferimento per gli errori di caricamento

Questa sezione contiene informazioni relative all'identificazione e alla risoluzione degli errori di caricamento dei dati.

ServiceException Errori S3

ServiceException Gli errori s3 più comuni sono causati da una stringa di credenziali formattata in modo errato o errato, dal fatto che il cluster e il bucket si trovano in AWS regioni diverse e autorizzazioni Amazon S3 insufficienti.

La sezione fornisce informazioni sulla risoluzione dei problemi per ogni tipo di errore.

Stringa di credenziali non valida

Se la stringa delle credenziali non è stata formattata correttamente, riceverai il seguente messaggio di errore:

```
ERROR: Invalid credentials. Must be of the format: credentials
'aws_access_key_id=<access-key-id>;aws_secret_access_key=<secret-access-key>
[;token=<temporary-session-token>]'
```

Verificare che la stringa di credenziali non contenga spazi o interruzioni di riga e sia racchiusa tra virgolette singole.

ID chiave di accesso non valido

Se l'ID della chiave di accesso non esiste, riceverai il seguente messaggio di errore:

```
[Amazon](500310) Invalid operation: S3ServiceException:The AWS Access Key Id you
 provided does not exist in our records.
```

Spesso si tratta di un errore di copia e incolla. Verifica che l'ID chiave di accesso sia stato inserito correttamente. Inoltre, se usi le chiavi di sessione temporanee, verifica che il valore per `token` sia impostato.

Chiave di accesso segreta non valida

Se la chiave di accesso segreta non è corretta, riceverai il seguente messaggio di errore:

```
[Amazon](500310) Invalid operation: S3ServiceException:The request signature we
 calculated does not match the signature you provided.
Check your key and signing method.,Status 403,Error SignatureDoesNotMatch
```

Spesso si tratta di un errore di copia e incolla. Verifica che la chiave di accesso segreta sia stata inserita correttamente e che sia la chiave corretta per l'ID chiave di accesso.

Bucket in una regione differente

Il bucket Amazon S3 specificato nel comando COPY deve trovarsi nella stessa AWS regione del cluster. Se il bucket Amazon S3 e il cluster si trovano in regioni differenti, verrà ricevuto un errore simile al seguente:

```
ERROR: S3ServiceException:The bucket you are attempting to access must be addressed
 using the specified endpoint.
```

È possibile creare un bucket Amazon S3 in una regione specifica selezionando la regione quando si crea il bucket con la console di gestione Amazon S3 oppure specificando un endpoint quando si crea il bucket con la CLI o l'API di Amazon S3. Per ulteriori informazioni, consulta Caricamento di file in Amazon S3.

Per maggiori informazioni sulle regioni di Amazon S3, consultare Accesso a un bucket nella Guida per l'utente di Amazon Simple Storage Service.

In alternativa, puoi specificare la regione utilizzando l'opzione REGION con il comando COPY.

Accesso negato

Se l'utente non dispone di autorizzazioni sufficienti, riceverai il seguente messaggio di errore:

```
ERROR: S3ServiceException:Access Denied,Status 403,Error AccessDenied
```

Una causa possibile è che l'utente identificato dalle credenziali non dispone dell'accesso LIST e GET al bucket Amazon S3. Per altre cause, consulta Risoluzione dei problemi relativi agli errori di accesso negato (403 Accesso negato) in Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Per informazioni sulla gestione dell'accesso utente ai bucket, consulta Identity and Access Management in Amazon S3 nella Guida per l'utente di Amazon Simple Storage.

Tabelle di sistema per la risoluzione di caricamento dei dati

Le seguenti tabelle di sistema Amazon Redshift possono essere utili nella risoluzione di problemi di caricamento dei dati:

- Esegui la query su STL_LOAD_ERRORS per scoprire gli errori che si sono verificati durante carichi specifici.
- Esegui la query su STL_FILE_SCAN per visualizzare i tempi di caricamento di file specifici o per vedere se un file specifico è stato letto.
- Eseguire la query su STL_S3CLIENT_ERROR per trovare i dettagli degli errori rilevati durante il trasferimento dei dati da Amazon S3.

Per trovare e diagnosticare errori di caricamento

1. Creare una vista o definire una query che restituisca dettagli sugli errori di caricamento. L'esempio seguente collega la tabella STL_LOAD_ERRORS alla tabella STV_TBL_PERM per fare corrispondere gli ID tabella con nomi tabella effettivi.

```
create view loadview as
(select distinct tbl, trim(name) as table_name, query, starttime,
trim(filename) as input, line_number, colname, err_code,
trim(err_reason) as reason
from stl_load_errors sl, stv_tbl_perm sp
where sl.tbl = sp.id);
```

2. Impostare l'opzione MAXERRORS nel comando COPY su un valore sufficientemente grande da consentire a COPY di restituire informazioni utili sui dati. Se COPY rileva errori, un messaggio di errore indica di consultare la tabella STL_LOAD_ERRORS per i dettagli.
3. Eseguire una query sulla vista LOADVIEW per visualizzare i dettagli dell'errore. Ad esempio:

```
select * from loadview where table_name='venue';
```

```
  tbl   | table_name | query |         starttime
--------+------------+-------+----------------------------
 100551 | venue      | 20974 | 2013-01-29 19:05:58.365391

|     input      | line_number | colname | err_code |       reason
+----------------+-------------+-------+----------+--------------------
| venue_pipe.txt |           1 |     0 |     1214 | Delimiter not found
```

4. Risolvere il problema nel file di input o nello script di caricamento, in base alle informazioni restituite dalla vista. Alcuni tipici errori di caricamento da tenere in considerazione includono:

- Mancata corrispondenza tra i tipi di dati nella tabella e i valori nei campi di dati di input.
- Mancata corrispondenza tra il numero di colonne nella tabella e il numero di campi nei dati di input.
- Virgolette non corrispondenti. Amazon Redshift supporta le virgolette singole e doppie; tuttavia, queste virgolette devono essere bilanciate in modo appropriato.
- Formato errato per dati di data/ora nei file di input.
- ut-of-range Valori O nei file di input (per colonne numeriche).
- Numero di valori distinti per una colonna superiore alla limitazione per la sua codifica di compressione.

Errori di caricamento di caratteri multibyte

Le colonne con un tipo di dati CHAR accettano solo caratteri UTF-8 a byte singolo, fino un valore di byte 127 o 7F esadecimale, che rappresenta anche il set di caratteri ASCII. Le colonne VARCHAR accettano caratteri multibyte UTF-8, fino a un massimo di quattro byte. Per ulteriori informazioni, consulta Tipi di carattere.

Se una riga nei dati di caricamento contiene un carattere non valido per il tipo di dati della colonna, COPY restituisce un errore e registra una riga nella tabella del log di sistema STL_LOAD_ERRORS con il numero di errore 1220. Il campo ERR_REASON include la sequenza di byte, in esadecimale, per il carattere non valido.

Un'alternativa alla correzione di caratteri non validi nei dati di caricamento è la sostituzione dei caratteri non validi durante il processo di caricamento. Per sostituire i caratteri UTF-8 non validi, specifica l'opzione ACCEPTINVCHARS con il comando COPY. Se l'opzione ACCEPTINVCHARS è impostata, il carattere specificato sostituisce il punto codice. Se l'opzione ACCEPTINVCHARS non è impostata, Amazon Redshift accetta i caratteri come UTF-8 valido. Per ulteriori informazioni, consulta ACCEPTINVCHARS.

Il seguente elenco di punti codice è valido UTF-8, le operazioni COPY non restituiscono un errore se l'opzione ACCEPTINVCHARS non è impostata. Tuttavia, questi punti codice sono caratteri non validi. Puoi utilizzare l'opzione ACCEPTINVCHARS per sostituire un punto di codice con un carattere specificato. Questi punti di codice includono l'intervallo di valori da `0xFDD0` a `0xFDEF` e valori fino a `0x10FFFF`, terminando con `FFFE` o `FFFF`:

- `0xFFFE`, `0x1FFFE`, `0x2FFFE`, …, `0xFFFFE`, `0x10FFFE`

- `0xFFFF`, `0x1FFFF`, `0x2FFFF`, ..., `0xFFFFF`, `0x10FFFF`

L'esempio seguente mostra il motivo dell'errore quando COPY prova a caricare il carattere UTF-8 `e0 a1 c7a4` in una colonna CHAR.

```
Multibyte character not supported for CHAR
(Hint: Try using  VARCHAR). Invalid char: e0 a1 c7a4
```

Se l'errore è correlato a un tipo di dati VARCHAR, il motivo dell'errore include un codice di errore e la sequenza esadecimale UTF-8 non valida. L'esempio seguente mostra il motivo dell'errore quando COPY prova a caricare il carattere UTF-8 `a4` in un campo VARCHAR.

```
String contains invalid or unsupported UTF-8 codepoints.
Bad UTF-8 hex sequence: a4 (error 3)
```

Nella tabella seguente sono elencate le descrizioni e le soluzioni alternative suggerite per gli errori di caricamento VARCHAR. Se si verifica uno di questi errori, sostituisci il carattere con una sequenza di codice UTF-8 valida o rimuovi il carattere.

Codice di errore	Descrizione
1	La sequenza di byte UTF-8 supera il massimo di quattro byte supportato da VARCHAR.
2	La sequenza di byte UTF-8 è incompleta. COPY non ha trovato il numero previsto di byte di continuazione per un carattere multibyte prima della fine della stringa.
3	Il carattere a byte singolo UTF-8 è fuori dell'intervallo. Il byte iniziale non deve essere 254, 255 o qualsiasi carattere compreso tra 128 e 191 (inclusi).
4	Il valore del byte finale nella sequenza di byte è fuori dell'intervallo. Il byte di continuazione deve essere compreso tra 128 e 191 (inclusi).
5	Il carattere UTF-8 è riservato come surrogato. I punti di codice surrogato (da U +D800 a U+DFFF) non sono validi.
8	La sequenza di byte supera il punto di codice UTF-8 massimo.

Codice di errore	Descrizione
9	La sequenza di byte UTF-8 non ha un punto di codice corrispondente.

Riferimento per gli errori di caricamento

Se si verificano errori durante il caricamento dei dati da un file, esegui una query sulla tabella STL_LOAD_ERRORS per identificare l'errore e determinare la possibile spiegazione. La seguente tabella elenca tutti i codici di errore che potrebbero verificarsi durante i caricamenti di dati:

Codici di errore di caricamento

Codice di errore	Descrizione
1200	Errore di analisi sconosciuto. Contatta il supporto.
1201	Il delimitatore di campo non è stato trovato nel file di input.
1202	I dati di input avevano più colonne di quelle definite nel DDL.
1203	I dati di input avevano meno colonne di quelle definite nel DDL.
1204	I dati di input hanno superato l'intervallo accettabile per il tipo di dati.
1205	Il formato della data non è valido. Consulta Stringhe DATEFORMAT e TIMEFORMAT per i formati validi.
1206	Il formato del timestamp non è valido. Consulta Stringhe DATEFORMAT e TIMEFORMAT per i formati validi.
1207	I dati contenevano un valore al di fuori dell'intervallo previsto di 0-9.
1208	Errore di formato del tipo di dati FLOAT.
1209	Errore di formato del tipo di dati DECIMAL.
1210	Errore di formato del tipo di dati BOOLEAN.

Codice di errore	Descrizione
1211	La riga di input non conteneva dati.
1212	Il file di caricamento non è stato trovato.
1213	Un campo specificato come NOT NULL non conteneva dati.
1214	Il delimitatore non è stato trovato.
1215	Errore del campo CHAR.
1.216	La riga di input non è valida.
1217	Il valore della colonna di identità non è valido.
1218	Quando utilizzavi NULL AS '\0', un campo contenente un terminatore null (NUL, or UTF-8 0000) conteneva più di un byte.
1219	L'esadecimale UTF-8 contiene una cifra non valida.
1220	La stringa contiene punti di codice UTF-8 non validi o non supportati.
1221	La codifica del file non è la stessa di quella specificata nel comando COPY.
1222	Errore di eccedenza del valore intero.
1223	Tipo di dati non valido.
1224	Dati di input in formato JSON o tipo di dati SUPER non ben formati.
8001	COPY con parametro MANIFEST richiede il percorso completo di un oggetto Amazon S3.
9005	Chiave finale specificata non valida.

Importazione continua di file da Amazon S3 (anteprima)

Questa è la documentazione non definitiva per la copia automatica (SQL COPY JOB), che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consiglia mo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. L'anteprima pubblica terminerà il 31 luglio 2024. I cluster di anteprima verranno rimossi automatic amente due settimane dopo il termine dell'anteprima. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

Note

È possibile creare un cluster Amazon Redshift di anteprima per testare nuove funzionalità di Amazon Redshift. Non è possibile utilizzare queste funzionalità in produzione o spostare il cluster di anteprima in un cluster di produzione o in un cluster su un'altra traccia. Per i termini e le condizioni dell'anteprima, consulta la sezione relativa a beta e anteprime nei termini del servizio AWS.

Come creare un cluster di anteprima

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegli Provisioned clusters dashboard (Pannello di controllo dei cluster con provisioning) e seleziona Clusters (Cluster). Regione AWS Sono elencati i cluster del tuo account nella versione corrente. Nelle colonne dell'elenco è visualizzato un sottoinsieme delle proprietà di ciascun cluster.
3. Nella pagina dell'elenco dei cluster viene visualizzato un banner che introduce l'anteprima. Scegli il pulsante Create preview cluster (Crea cluster di anteprima) per aprire la pagina di creazione del cluster.
4. Inserisci le proprietà del cluster. Scegli la traccia di anteprima che contiene le funzionalità che desideri testare. Consigliamo di assegnare al cluster un nome che indichi che si trova in una traccia di anteprima. Scegli le opzioni per il cluster, tra cui quelle contrassegnate come -anteprima, per le funzionalità che desideri testare. Per informazioni generali sulla creazione di cluster, consulta Creazione di un cluster nella Guida alla gestione di Amazon Redshift.
5. Per creare un cluster di anteprima, scegli Crea cluster.

6. Quando il cluster di anteprima è disponibile, utilizza il client SQL per caricare ed eseguire query sui dati.

È necessario creare il cluster con la traccia di anteprima denominata:`preview_2023`. Per i test è necessario creare un nuovo cluster; il ripristino di un cluster in questa traccia non è supportato. La funzionalità di copia automatica non è disponibile per il gruppo di lavoro Amazon Redshift serverless.
Questa anteprima è disponibile nei seguenti formati: Regioni AWS

- Regione Stati Uniti orientali (Ohio) (us-east-2)
- Regione Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Regione Stati Uniti occidentali (Oregon) (us-west-2)
- Regione Asia Pacifico (Tokyo) (ap-northeast-1)
- Regione Europa (Stoccolma) (eu-north-1)
- Regione Europa (Irlanda) (eu-west-1)

È possibile utilizzare un processo COPY JOB per caricare dati nelle tabelle Amazon Redshift da file archiviati in Amazon S3. Amazon Redshift rileva quando vengono aggiunti nuovi file Amazon S3 al percorso specificato nel comando COPY ed esegue automaticamente un comando COPY senza che sia necessario creare una pipeline di importazione dei dati esterna. Amazon Redshift consente di tenere traccia in modo semplice dei file caricati. Amazon Redshift stabilisce il numero di file raggruppati per comando COPY. È possibile visualizzare i comandi COPY risultanti nelle viste di sistema.

È necessario definire un processo COPY JOB una sola volta, in quanto gli stessi parametri verranno utilizzati per le esecuzioni future.

È possibile gestire le operazioni di caricamento utilizzando le opzioni CREATE, LIST, SHOW, DROP, ALTER e RUN relative ai processi. Per ulteriori informazioni, consulta COPY JOB (anteprima).

È possibile eseguire query sulle viste di sistema per vedere lo stato e l'avanzamento del processo COPY JOB. Le viste disponibili sono le seguenti:

- SYS_COPY_JOB (anteprima) – include una riga per ogni processo COPY JOB definito.
- STL_LOAD_ERRORS – include gli errori relativi ai comandi COPY.

- STL_LOAD_COMMITS – include le informazioni utilizzate per risolvere i problemi di caricamento dei dati relativi al comando COPY.
- SYS_LOAD_HISTORY – include i dettagli sui comandi COPY.
- SYS_LOAD_ERROR_DETAIL – include i dettagli sugli errori relativi al comando COPY.

Per ottenere l'elenco dei file caricati da un processo COPY JOB, esegui l'esempio qui sotto sostituendo *<job_id>*:

```
SELECT job_id, job_name, data_source, copy_query,filename,status, curtime
FROM sys_copy_job copyjob
JOIN stl_load_commits loadcommit
ON copyjob.job_id = loadcommit.copy_job_id
WHERE job_id = <job_id>;
```

Aggiornamento di tabelle con comandi DML

Amazon Redshift supporta i comandi DML (Data Manipulation Language) standard (INSERT, UPDATE e DELETE) che è possibile utilizzare per modificare le righe nelle tabelle. È inoltre possibile utilizzare il comando TRUNCATE per eseguire eliminazioni di massa rapide.

> **Note**
>
> Consigliamo l'uso del comando COPY per caricare grandi quantità di dati. L'utilizzo di singole istruzioni INSERT per popolare una tabella potrebbe essere eccessivamente lento. In alternativa, se i dati sono già presenti in altre tabelle di database Amazon Redshift, utilizzare INSERT INTO... SELECT FROM o CREATE TABLE AS per migliorare le prestazioni. Per informazioni, consulta INSERT or CREATE TABLE AS.

Se inserisci, aggiorni o elimini un numero significativo di righe in una tabella, rispetto al numero di righe prima delle modifiche, al termine dell'operazione, esegui i comandi ANALYZE e VACUUM nella tabella. Se nell'applicazione si accumula una serie di piccole modifiche, potresti voler programmare l'esecuzione a intervalli regolari dei comandi ANALYZE e VACUUM. Per ulteriori informazioni, consulta Analisi delle tabelle e Vacuum delle tabelle.

Aggiornamento e inserimento di nuovi dati

È possibile aggiungere in modo efficiente nuovi dati a una tabella esistente utilizzando il comando MERGE. Eseguite un'operazione di unione creando una tabella intermedia e quindi utilizzando uno dei metodi descritti in questa sezione per aggiornare la tabella di destinazione dalla tabella intermedia. Per ulteriori informazioni sul comando MERGE, consulta MERGE.

Argomenti

Gli Esempi di unione utilizzano un set di dati di esempio per Amazon Redshift, chiamato TICKIT. Come prerequisito, puoi configurare le tabelle e i dati TICKIT seguendo le istruzioni disponibili in Nozioni di base sulle attività di database comuni. Informazioni più dettagliate sul set di dati di esempio sono disponibili in Database di esempio.

Metodo di unione 1: sostituzione di righe esistenti

Se si sovrascrivono tutte le colonne della tabella di destinazione, il metodo più veloce per eseguire un'unione consiste nel sostituire le righe esistenti. Questa operazione effettua la scansione della tabella di destinazione solo una volta, utilizzando un inner join per eliminare le righe che verranno aggiornate. Una volta eliminate le righe, vengono sostituite insieme alle nuove righe da una singola operazione INSERT dalla tabella di gestione temporanea.

Utilizza questo metodo se tutte le seguenti condizioni sono vere:

- La tabella di destinazione e la tabella di gestione temporanea contengono le stesse colonne.
- Intendi sostituire tutti i dati nelle colonne della tabella di destinazione con tutte le colonne della tabella di gestione temporanea.
- Utilizzerai tutte le righe nella tabella di gestione temporanea nell'unione.

Se uno qualsiasi di questi criteri non si applica, utilizza il Metodo di unione 2: specifica di un elenco di colonne con MERGE, descritto nella sezione seguente.

Se non utilizzi tutte le righe nella tabella di gestione temporanea, è possibile filtrare le istruzioni DELETE e INSERT tramite una clausola WHERE per escludere le righe di cui non è effettivamente in corso la modifica. Tuttavia, se la maggior parte delle righe nella tabella di gestione temporanea non verrà inclusa nell'unione, consigliamo di eseguire un UPDATE e un INSERT in fasi separate, come descritto più avanti in questa sezione.

Metodo di unione 2: specifica di un elenco di colonne senza usare MERGE

Utilizza questo metodo per aggiornare colonne specifiche nella tabella di destinazione anziché sovrascrivere intere righe. Questo metodo richiede più tempo del metodo precedente perché richiede una fase di aggiornamento aggiuntiva e non usa il comando MERGE. Utilizza questo metodo se una delle seguenti condizioni è vera:

- Non tutte le colonne nella tabella di destinazione devono essere aggiornate.
- La maggior parte delle righe nella tabella di gestione temporanea non verrà utilizzata negli aggiornamenti.

Creazione di una tabella di gestione temporanea

La tabella di gestione temporanea è una tabella temporanea che contiene tutti i dati che verranno utilizzati per apportare modifiche alla tabella di destinazione, inclusi gli aggiornamenti e gli inserimenti.

Un'operazione di unione richiede un join tra la tabella di gestione temporanea e la tabella di destinazione. Per collocare le righe di unione, imposta la chiave di distribuzione della tabella di gestione temporanea sulla stessa colonna della chiave di distribuzione della tabella di destinazione. Ad esempio, se la tabella di destinazione utilizza una colonna chiave esterna come chiave di distribuzione, utilizza la stessa colonna per la chiave di distribuzione della tabella di gestione temporanea. Se crei una tabella di gestione temporanea tramite un'istruzione CREATE TABLE LIKE, la tabella di gestione temporanea erediterà la chiave di distribuzione dalla tabella padre. Se utilizzi un'istruzione CREATE TABLE AS, la nuova tabella non eredita la chiave di distribuzione. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati

Se la chiave di distribuzione non è uguale alla chiave primaria e la chiave di distribuzione non viene aggiornata come parte dell'operazione di unione, aggiungi un predicato di join ridondante alle colonne della chiave di distribuzione per abilitare un join collocato. Ad esempio:

```
where target.primarykey = stage.primarykey
and target.distkey = stage.distkey
```

Per verificare che la query utilizzerà un join collocato, esegui la query con EXPLAIN e controlla DS_DIST_NONE su tutti i join. Per ulteriori informazioni, consulta Valutazione del piano di query

Esecuzione di un'operazione di unione tramite sostituzione di righe esistenti

Quando esegui l'operazione di unione descritta nella procedura, inserisci tutti i passaggi, tranne la creazione e l'eliminazione della tabella di gestione temporanea, in un'unica transazione. Verrà eseguito il rollback della transazione se una fase restituisce un errore. L'utilizzo di una singola transazione riduce anche il numero di commit, risparmiando tempo e risorse.

Per eseguire un'operazione di unione tramite sostituzione di righe esistenti

1. Creare una tabella di gestione temporanea e quindi popolarla con i dati da unire, come mostrato nel seguente pseudocodice.

   ```
   create temp table stage (like target);

   insert into stage
   select * from source
   where source.filter = 'filter_expression';
   ```

2. Utilizzate MERGE per eseguire un'unione interna con la tabella intermedia per aggiornare le righe della tabella di destinazione che corrispondono alla tabella intermedia, quindi inserite tutte le righe rimanenti nella tabella di destinazione che non corrispondono alla tabella intermedia.

   Ti consigliamo di eseguire le operazioni di aggiornamento e inserimento in un unico comando MERGE.

   ```
   MERGE INTO target
   USING stage [optional alias] on (target.primary_key = stage.primary_key)
   WHEN MATCHED THEN
   UPDATE SET col_name1 = stage.col_name1 , col_name2= stage.col_name2, col_name3 =
    {expr}
   WHEN NOT MATCHED THEN
   ```

```
INSERT (col_name1 , col_name2, col_name3) VALUES (stage.col_name1, stage.col_name2,
 {expr});
```

3. Rilasciare la tabella di gestione temporanea.

```
drop table stage;
```

Esecuzione di un'operazione di unione tramite specificazione di un elenco di colonne senza l'uso del comando MERGE

Quando esegui l'operazione di unione descritta nella procedura, inserisci tutti i passaggi in un'unica transazione. Verrà eseguito il rollback della transazione se una fase restituisce un errore. L'utilizzo di una singola transazione riduce anche il numero di commit, risparmiando tempo e risorse.

Per eseguire un'operazione di unione tramite specificazione di un elenco di colonne

1. Inserisci l'intera operazione in un unico blocco di transazioni.

```
begin transaction;
…
end transaction;
```

2. Creare una tabella di gestione temporanea e quindi popolarla con i dati da unire, come mostrato nel seguente pseudocodice.

```
create temp table stage (like target);
insert into stage
select * from source
where source.filter = 'filter_expression';
```

3. Aggiornare la tabella di destinazione tramite un inner join con la tabella di gestione temporanea.

   - Nella clausola UPDATE, elencare in modo esplicito le colonne da aggiornare.
   - Eseguire un inner join con la tabella di gestione temporanea.
   - Se la chiave di distribuzione è diversa dalla chiave primaria e la chiave di distribuzione non viene aggiornata, aggiungere un predicato di join ridondante alle colonne della chiave di distribuzione per abilitare un join collocato. Per verificare che la query utilizzerà un join collocato, esegui la query con EXPLAIN e controlla DS_DIST_NONE su tutti i join. Per ulteriori informazioni, consulta Valutazione del piano di query

- Se la tabella di destinazione è ordinata per timestamp, aggiungere un predicato per sfruttare le scansioni a intervallo limitato nella tabella di destinazione. Per ulteriori informazioni, consulta Best practice di Amazon Redshift per la progettazione di query.
- Se non utilizzi tutte le righe nell'unione, aggiungi una clausola per filtrare le righe che desideri modificare. Ad esempio, aggiungere un filtro disuguaglianza a una o più colonne per escludere le righe che non sono state modificate.
- Inserire le operazioni di aggiornamento, eliminazione e inserimento in un singolo blocco di transazione, in modo che in caso di problemi venga eseguito il rollback.

Ad esempio:

```
begin transaction;

update target
set col1 = stage.col1,
col2 = stage.col2,
col3 = 'expression'
from stage
where target.primarykey = stage.primarykey
and target.distkey = stage.distkey
and target.col3 > 'last_update_time'
and (target.col1 != stage.col1
or target.col2 != stage.col2
or target.col3 = 'filter_expression');
```

4. Eliminare le righe non necessarie dalla tabella di gestione temporanea tramite un inner join con la tabella di destinazione. Alcune righe nella tabella di destinazione corrispondono già alle righe corrispondenti nella tabella di gestione temporanea e altre sono state aggiornate nella fase precedente. In entrambi i casi, non sono necessarie per l'inserimento.

```
delete from stage
using target
where stage.primarykey = target.primarykey;
```

5. Inserire tutte le righe rimanenti dalla tabella di gestione temporanea. Utilizzare lo stesso elenco di colonne nella clausola VALUES utilizzato nell'istruzione UPDATE nella fase due.

```
insert into target
(select col1, col2, 'expression'
```

```
from stage);

end transaction;
```

6. Rilasciare la tabella di gestione temporanea.

```
drop table stage;
```

Esempi di unione

Negli esempi seguenti viene eseguita un'unione per aggiornare la tabella SALES. Il primo esempio utilizza il metodo più semplice di eliminazione dalla tabella di destinazione e quindi l'inserimento di tutte le righe dalla tabella di gestione temporanea. La seconda fase richiede l'aggiornamento su colonne selezionate nella tabella di destinazione, quindi include una fase di aggiornamento aggiuntiva.

Gli Esempi di unione utilizzano un set di dati di esempio per Amazon Redshift, chiamato TICKIT. Come prerequisito, puoi configurare le tabelle e i dati TICKIT seguendo le istruzioni disponibili in Nozioni di base sulle attività di database comuni. Informazioni più dettagliate sul set di dati di esempio sono disponibili in Database di esempio.

Origine dati di unione di esempio

Gli esempi in questa sezione richiedono un'origine dati di esempio che includa sia gli aggiornamenti sia gli inserimenti. Per gli esempi, creeremo una tabella di esempio denominata SALES_UPDATE che utilizza i dati dalla tabella SALES. Popoleremo la nuova tabella con dati casuali che rappresentano nuove attività di vendita per dicembre. Utilizzeremo la tabella di esempio SALES_UPDATE per creare la tabella di gestione temporanea negli esempi che seguono.

```
-- Create a sample table as a copy of the SALES table.

create table tickit.sales_update as
select * from tickit.sales;

-- Change every fifth row to have updates.

update tickit.sales_update
set qtysold = qtysold*2,
pricepaid = pricepaid*0.8,
commission = commission*1.1
```

```
where saletime > '2008-11-30'
and mod(sellerid, 5) = 0;

-- Add some new rows to have inserts.
-- This example creates a duplicate of every fourth row.

insert into tickit.sales_update
select (salesid + 172456) as salesid, listid, sellerid, buyerid, eventid, dateid,
 qtysold, pricepaid, commission, getdate() as saletime
from tickit.sales_update
where saletime > '2008-11-30'
and mod(sellerid, 4) = 0;
```

Esempio di un'unione che sostituisce le righe esistenti in base alle chiavi di corrispondenza

Il seguente script utilizza la tabella SALES_UPDATE per eseguire un'operazione di unione nella tabella SALES con nuovi dati per l'attività di vendita di dicembre. Questo esempio sostituisce le righe della tabella SALES che contengono aggiornamenti. Per questo esempio, vogliamo aggiornare le colonne QTYSOLD e PRICEPAID e non modificare COMMISSION e SALETIME.

```
MERGE into tickit.sales
USING tickit.sales_update sales_update
on ( sales.salesid = sales_update.salesid
and sales.listid = sales_update.listid
and sales_update.saletime > '2008-11-30'
and (sales.qtysold != sales_update.qtysold
or sales.pricepaid != sales_update.pricepaid))
WHEN MATCHED THEN
update SET qtysold = sales_update.qtysold,
pricepaid = sales_update.pricepaid
WHEN NOT MATCHED THEN
INSERT (salesid, listid, sellerid, buyerid, eventid, dateid, qtysold , pricepaid,
 commission, saletime)
values (sales_update.salesid, sales_update.listid, sales_update.sellerid,
 sales_update.buyerid, sales_update.eventid,
sales_update.dateid, sales_update.qtysold , sales_update.pricepaid,
 sales_update.commission, sales_update.saletime);

-- Drop the staging table.
drop table tickit.sales_update;

-- Test to see that commission and salestime were not impacted.
```

```
SELECT sales.salesid, sales.commission, sales.salestime, sales_update.commission,
 sales_update.salestime
FROM tickit.sales
INNER JOIN tickit.sales_update sales_update
ON
sales.salesid = sales_update.salesid
AND sales.listid = sales_update.listid
AND sales_update.saletime > '2008-11-30'
AND (sales.commission != sales_update.commission
OR sales.salestime != sales_update.salestime);
```

Esempio di un'unione che specifica un elenco di colonne senza usare MERGE

L'esempio seguente esegue un'operazione di unione per aggiornare SALES con i nuovi dati per l'attività di vendita di dicembre. Sono necessari dati di esempio che includano sia gli aggiornamenti sia gli inserimenti, oltre alle righe che non sono state modificate. Per questo esempio, vogliamo aggiornare le colonne QTYSOLD e PRICEPAID e non modificare COMMISSION e SALETIME. Il seguente script utilizza la tabella SALES_UPDATE per eseguire un'operazione di unione nella tabella SALES.

```
-- Create a staging table and populate it with rows from SALES_UPDATE for Dec
create temp table stagesales as select * from sales_update
where saletime > '2008-11-30';

-- Start a new transaction
begin transaction;

-- Update the target table using an inner join with the staging table
-- The join includes a redundant predicate to collocate on the distribution key -- A
 filter on saletime enables a range-restricted scan on SALES

update sales
set qtysold = stagesales.qtysold,
pricepaid = stagesales.pricepaid
from stagesales
where sales.salesid = stagesales.salesid
and sales.listid = stagesales.listid
and stagesales.saletime > '2008-11-30'
and (sales.qtysold != stagesales.qtysold
or sales.pricepaid != stagesales.pricepaid);

-- Delete matching rows from the staging table
-- using an inner join with the target table
```

```
delete from stagesales
using sales
where sales.salesid = stagesales.salesid
and sales.listid = stagesales.listid;

-- Insert the remaining rows from the staging table into the target table
insert into sales
select * from stagesales;

-- End transaction and commit
end transaction;

-- Drop the staging table
drop table stagesales;
```

Esecuzione di una copia completa

Una copia completa ricrea e ripopola una tabella tramite inserimento di massa, che ordina automaticamente la tabella. Se una tabella ha una regione ampia e non ordinata, una copia completa è molto più veloce di un vacuum. Ti consigliamo di effettuare aggiornamenti simultanei durante un'operazione di copia completa solo se riesci a tenerne traccia. Una volta completato il processo, sposta gli aggiornamenti delta nella nuova tabella. Un'operazione VACUUM supporta automaticamente gli aggiornamenti simultanei.

È possibile scegliere uno dei seguenti metodi per creare una copia della tabella originale:

- Utilizza la tabella con DDL originale.

  Se è disponibile CREATE TABLE DDL, questo è il metodo più veloce e preferito. Se crei una nuova tabella, è possibile specificare tutti gli attributi di tabella e colonna, comprese la chiave primaria e le chiavi esterne. È possibile trovare il DDL originale utilizzando la funzione SHOW TABLE.
- Utilizza CREATE TABLE LIKE.

  Se il DDL originale non è disponibile, è possibile utilizzare CREATE TABLE LIKE per ricreare la tabella originale. La nuova tabella eredita la codifica, la chiave di distribuzione, la chiave di ordinamento e gli attributi non null della tabella padre. La nuova tabella non eredita la chiave primaria e gli attributi di chiave esterna della tabella padre, ma è possibile aggiungerli utilizzando ALTER TABLE.
- Crea una tabella temporanea e tronca la tabella originale.

Se è necessario mantenere gli attributi della chiave primaria e della chiave esterna della tabella principale. Se la tabella principale ha dipendenze, puoi usare CREATE TABLE ... AS (CTAS) per creare una tabella temporanea. Quindi troncare la tabella originale e popolarla dalla tabella temporanea.

L'utilizzo di una tabella temporanea migliora le prestazioni in modo significativo rispetto all'utilizzo di una tabella permanente, ma c'è il rischio di perdere dati. La tabella temporanea viene automaticamente eliminata alla fine della sessione in cui è stata creata. TRUNCATE viene sottoposto immediatamente al commit, anche se si trova all'interno di un blocco di transazione. Se TRUNCATE ha esito positivo ma la sessione termina prima del completamento del comando INSERT successivo, i dati vengono persi. Se la perdita di dati non è accettabile, utilizza una tabella permanente.

Dopo aver creato una copia di una tabella, potrebbe essere necessario concedere l'accesso alla nuova tabella. È possibile utilizzare GRANT per definire i privilegi di accesso. Per visualizzare e concedere tutti i privilegi di accesso di una tabella, devi avere uno dei seguenti ruoli:

- Un utente con privilegi avanzati.
- Il proprietario della tabella che si desidera copiare.
- Un utente con il privilegio ACCESS SYSTEM TABLE per visualizzare i privilegi della tabella e con il privilegio di concessione per tutte le autorizzazioni pertinenti.

Inoltre, potrebbe essere necessario concedere l'autorizzazione all'utilizzo dello schema in cui si trova la copia completa. È necessario concedere l'autorizzazione all'utilizzo se lo schema della copia completa è diverso dallo schema della tabella originale e inoltre non è lo schema `public`. Per visualizzare e concedere tutti i privilegi di utilizzo, devi avere uno dei seguenti ruoli:

- Un utente con privilegi avanzati.
- Un utente che può concedere l'autorizzazione USAGE per lo schema della copia completa.

Per eseguire una copia completa utilizzando la tabella con DDL originale

1. (Facoltativo) Ricreare la tabella DDL eseguendo uno script chiamato `v_generate_tbl_ddl`.
2. Creare una copia della tabella utilizzando l'originale CREATE TABLE DDL.

3. Utilizzare un'istruzione INSERT INTO ... SELECT per popolare la copia con i dati della tabella originale.
4. Verifica le autorizzazioni concesse nella tabella precedente. Puoi visualizzare queste autorizzazioni nella vista di sistema SVV_RELATION_PRIVILEGES.
5. Se necessario, concedi le autorizzazioni della tabella precedente alla nuova tabella.
6. Concedi l'autorizzazione all'utilizzo a tutti i gruppi e utenti con privilegi nella tabella originale. Questo passaggio non è necessario se la tabella di copia completa è nello schema `public` o nello stesso schema della tabella originale.
7. Rilasciare la tabella originale.
8. Utilizzare un'istruzione ALTER TABLE per assegnare un nuovo nome alla copia con il nome della tabella originale.

L'esempio seguente esegue una copia completa sulla tabella SAMPLE utilizzando un duplicato di SAMPLE denominato sample_copy.

```
--Create a copy of the original table in the sample_namespace namespace using the
 original CREATE TABLE DDL.
create table sample_namespace.sample_copy ( … );

--Populate the copy with data from the original table in the public namespace.
insert into sample_namespace.sample_copy (select * from public.sample);

--Check SVV_RELATION_PRIVILEGES for the original table's privileges.
select * from svv_relation_privileges where namespace_name = 'public' and relation_name
 = 'sample' order by identity_type, identity_id, privilege_type;

--Grant the original table's privileges to the copy table.
grant DELETE on table sample_namespace.sample_copy to group group1;
grant INSERT, UPDATE on table sample_namespace.sample_copy to group group2;
grant SELECT on table sample_namespace.sample_copy to user1;
grant INSERT, SELECT, UPDATE on table sample_namespace.sample_copy to user2;

--Grant usage permission to every group and user that has privileges in the original
 table.
grant USAGE on schema sample_namespace to group group1, group group2, user1, user2;

--Drop the original table.
drop table public.sample;
```

```
--Rename the copy table to match the original table's name.
alter table sample_namespace.sample_copy rename to sample;
```

Per eseguire una copia completa utilizzando CREATE TABLE LIKE

1. Creare un nuovo tabella utilizzando CREATE TABLE LIKE.
2. Utilizzare un'istruzione INSERT INTO ... SELECT per copiare le righe dalla tabella attuale nella nuova tabella.
3. Verifica le autorizzazioni concesse nella tabella precedente. Puoi visualizzare queste autorizzazioni nella vista di sistema SVV_RELATION_PRIVILEGES.
4. Se necessario, concedi le autorizzazioni della tabella precedente alla nuova tabella.
5. Concedi l'autorizzazione all'utilizzo a tutti i gruppi e utenti con privilegi nella tabella originale. Questo passaggio non è necessario se la tabella di copia completa è nello schema `public` o nello stesso schema della tabella originale.
6. Rilasciare la tabella attuale.
7. Utilizzare un'istruzione ALTER TABLE per assegnare un nuovo nome alla nuova tabella con il nome della tabella originale.

L'esempio seguente esegue una copia completa sulla tabella SAMPLE utilizzando CREATE TABLE LIKE.

```
--Create a copy of the original table in the sample_namespace namespace using CREATE
 TABLE LIKE.
create table sameple_namespace.sample_copy (like public.sample);

--Populate the copy with data from the original table.
insert into sample_namespace.sample_copy (select * from public.sample);

--Check SVV_RELATION_PRIVILEGES for the original table's privileges.
select * from svv_relation_privileges where namespace_name = 'public' and relation_name
 = 'sample' order by identity_type, identity_id, privilege_type;

--Grant the original table's privileges to the copy table.
grant DELETE on table sample_namespace.sample_copy to group group1;
grant INSERT, UPDATE on table sample_namespace.sample_copy to group group2;
grant SELECT on table sample_namespace.sample_copy to user1;
grant INSERT, SELECT, UPDATE on table sample_namespace.sample_copy to user2;
```

```
--Grant usage permission to every group and user that has privileges in the original
 table.
grant USAGE on schema sample_namespace to group group1, group group2, user1, user2;

--Drop the original table.
drop table public.sample;

--Rename the copy table to match the original table's name.
alter table sample_namespace.sample_copy rename to sample;
```

Per eseguire una copia completa tramite la creazione di una tabella temporanea e il troncamento della tabella originale

1. Utilizzare CREATE TABLE AS per creare una tabella temporanea con le righe della tabella originale.
2. Troncare la tabella attuale.
3. Utilizzare un'istruzione INSERT INTO ... SELECT per copiare le righe dalla tabella temporanea nella tabella originale.
4. Rilasciare la tabella temporanea.

L'esempio seguente esegue una copia completa sulla tabella SALES tramite la creazione di una tabella temporanea e il troncamento della tabella originale. Poiché la tabella originale resta, non è necessario concedere autorizzazioni alla tabella di copia.

```
--Create a temp table copy using CREATE TABLE AS.
create temp table salestemp as select * from sales;

--Truncate the original table.
truncate sales;

--Copy the rows from the temporary table to the original table.
insert into sales (select * from salestemp);

--Drop the temporary table.
drop table salestemp;
```

Analisi delle tabelle

L'operazione ANALYZE aggiorna i metadati statistici utilizzati dal pianificatore di query per scegliere i piani ottimali.

Nella maggior parte dei casi, non è necessario eseguire esplicitamente il comando ANALYZE. Amazon Redshift monitora le modifiche al carico di lavoro e aggiorna automaticamente le statistiche in background. Inoltre, il comando COPY esegue automaticamente un'analisi quando carica i dati in una tabella vuota.

Per analizzare in modo esplicito una tabella o l'intero database, esegui il comando ANALYZE.

Argomenti

Analisi automatica

Amazon Redshift monitora ininterrottamente il database ed esegue automaticamente le operazioni di analisi in background. Per ridurre l'impatto sulle prestazioni del sistema, l'analisi automatica viene eseguita quando i carichi di lavoro sono più leggeri.

L'analisi automatica è abilitata per impostazione predefinita. Per disattivare l'analisi automatica, imposta il parametro `auto_analyze` su **`false`** modificando il gruppo di parametri del cluster.

Per ridurre il tempo di elaborazione e migliorare le prestazioni generali del sistema, Amazon Redshift ignora l'analisi automatica per qualsiasi tabella che abbia una percentuale di modifiche limitata.

Un'operazione di analisi ignora le tabelle che contengono up-to-date statistiche. Se esegui ANALYZE nel flusso di lavoro ETL (Extract, Transform and Load, estrazione, trasformazione e caricamento), l'analisi automatica ignora le tabelle con statistiche aggiornate. Analogamente, un comando ANALYZE esplicito ignora le tabelle se l'analisi automatica ha aggiornato le statistiche della tabella.

Analisi dei dati di nuove tabelle

Per impostazione predefinita, il comando COPY esegue un comando ANALYZE dopo aver caricato i dati in una tabella vuota. Puoi forzare un comando ANALYZE indipendentemente dal fatto che una tabella sia vuota impostando STATUPDATE ON. Se specifichi STATUPDATE OFF, ANALYZE non

viene eseguito. Solo il proprietario della tabella o un utente con privilegi avanzati può eseguire il comando ANALYZE o eseguire il comando COPY con STATUPDATE impostata su ON.

Amazon Redshift analizza anche le nuove tabelle create con i seguenti comandi:

- CREATE TABLE AS (CTAS)
- CREATE TEMP TABLE AS
- SELECT INTO

Auando si esegue una query su una nuova tabella che non è stata analizzata dopo il caricamento iniziale dei dati, Amazon Redshift restituisce un messaggio di avviso. Non vengono visualizzati avvisi quando esegui una query su una tabella dopo un successivo aggiornamento o caricamento. Lo stesso messaggio di avviso viene restituito quando esegui il comando EXPLAIN su una query che fa riferimento a tabelle non analizzate.

Se l'aggiunta di dati a una tabella non vuota modifica significativamente le dimensioni della tabella, puoi aggiornare le statistiche in modo esplicito. Per farlo, esegui il comando ANALYZE o l'opzione STATUPDATE ON con il comando COPY. Per visualizzare i dettagli sul numero di righe che sono state inserite o eliminate dall'ultimo ANALYZE, esegui una query sulla tabella del catalogo di sistema PG_STATISTIC_INDICATOR.

Puoi specificare l'ambito del comando ANALYZE su una delle opzioni seguenti:

- L'intero database attuale
- Una tabella singola
- Una o più colonne specifiche in una singola tabella
- Colonne che possono essere utilizzate come predicati nelle query

Il comando ANALYZE ottiene un campione di righe dalla tabella, esegue alcuni calcoli e salva le statistiche delle colonne risultanti. Per impostazione predefinita, Amazon Redshift esegue un passaggio di esempio per la colonna DISTKEY e un altro passaggio di esempio per tutte le altre colonne nella tabella. Se desideri generare statistiche per un sottoinsieme di colonne, è possibile specificare un elenco di colonne separate da virgola. Puoi eseguire ANALYZE con la clausola PREDICATE COLUMNS per ignorare le colonne non utilizzate come predicati.

Le operazioni di ANALYZE richiedono molte risorse, quindi eseguile solo su tabelle e colonne che richiedono effettivamente aggiornamenti delle statistiche. Non è necessario analizzare tutte le

colonne in tutte le tabelle regolarmente o nella stessa pianificazione. Se i dati cambiano in modo sostanziale, analizza le colonne utilizzate frequentemente nel modo seguente:

- Operazioni di raggruppamento e ordinamento
- Join
- Predicati di query

Per ridurre i tempi di elaborazione e migliorare le prestazioni generali del sistema, Amazon Redshift ignora ANALYZE per qualsiasi tabella che abbia una bassa percentuale di righe modificate, come determinato dal parametro analyze_threshold_percent. Per impostazione predefinita, la soglia di analisi è impostata su 10 percento. È possibile modificare la soglia di analisi per la sessione attuale eseguendo un comando SET.

Le colonne che hanno meno probabilità di richiedere analisi frequenti sono quelle che rappresentano fatti e misure e tutti gli attributi correlati su cui non viene mai effettivamente eseguita una query, come le colonne VARCHAR di grandi dimensioni. Ad esempio, prendi in considerazione la tabella LISTING nel database TICKIT.

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'listing';

column         |        type        | encoding | distkey | sortkey
---------------+--------------------+----------+---------+---------
listid         | integer            | none     | t       | 1
sellerid       | integer            | none     | f       | 0
eventid        | integer            | mostly16 | f       | 0
dateid         | smallint           | none     | f       | 0
numtickets     | smallint           | mostly8  | f       | 0
priceperticket | numeric(8,2)       | bytedict | f       | 0
totalprice     | numeric(8,2)       | mostly32 | f       | 0
listtime       | timestamp with...  | none     | f       | 0
```

Se questa tabella viene caricata ogni giorno con un numero elevato di nuovi record, la colonna LISTID, che viene spesso utilizzata nelle query come chiave di join, deve essere analizzata regolarmente. Se TOTALPRICE e LISTTIME sono i vincoli utilizzati frequentemente nelle query, è possibile analizzare tali colonne e la chiave di distribuzione in ogni giorno della settimana.

```
analyze listing(listid, totalprice, listtime);
```

Se i venditori e gli eventi nell'applicazione sono molto più statici e gli ID di data si riferiscono a un set fisso di giorni che copre solo due o tre anni, i valori univoci per queste colonne non cambiano significativamente. Tuttavia, il numero di istanze di ogni valore univoco aumenterà costantemente.

Inoltre, prendi in considerazione il caso in cui venga eseguita raramente una query sulle misure NUMTICKETS e PRICEPERTICKET rispetto alla colonna TOTALPRICE. In questo caso, puoi eseguire il comando ANALYZE sull'intera tabella una volta ogni fine settimana per aggiornare le statistiche relative alle cinque colonne non analizzate giornalmente:

Colonne di predicato

Come valida alternativa alla specifica di un elenco di colonne, è possibile scegliere di analizzare solo le colonne che potrebbero essere utilizzate come predicati. Quando si esegue una query, tutte le colonne utilizzate in un join, una condizione del filtro o una clausola di raggruppamento vengono contrassegnate come colonne di predicato nel catalogo di sistema. Quando esegui ANALYZE con la clausola PREDICATE COLUMNS, l'operazione di analisi include solo le colonne che soddisfano i seguenti criteri:

- La colonna è contrassegnata come una colonna di predicato.
- La colonna è una chiave di distribuzione.
- La colonna fa parte di una chiave di ordinamento.

Se nessuna delle colonne di una tabella è contrassegnata come predicato, ANALYZE include tutte le colonne, anche quando viene specificato PREDICATE COLUMNS. Se nessuna colonna è contrassegnata come colonne di predicato, potrebbe essere perché non è stata ancora eseguita una query sulla tabella.

È possibile scegliere di utilizzare PREDICATE COLUMNS quando il modello di query del carico di lavoro è relativamente stabile. Quando il modello di query è variabile, con colonne diverse utilizzate frequentemente come predicati, l'uso di PREDICATE COLUMNS potrebbe temporaneamente portare a statistiche obsolete. Le statistiche obsolete possono portare a piani di runtime delle query non ottimali e tempi di runtime lunghi. Tuttavia, la volta successiva che esegui ANALYZE utilizzando PREDICATE COLUMNS, vengono incluse le nuove colonne del predicato.

Per visualizzare i dettagli delle colonne dei predicati, utilizza il seguente codice SQL per creare una vista denominata PREDICATE_COLUMNS.

```
CREATE VIEW predicate_columns AS
WITH predicate_column_info as (
```

```
SELECT ns.nspname AS schema_name, c.relname AS table_name, a.attnum as col_num,
 a.attname as col_name,
        CASE
            WHEN 10002 = s.stakind1 THEN array_to_string(stavalues1, '||')
            WHEN 10002 = s.stakind2 THEN array_to_string(stavalues2, '||')
            WHEN 10002 = s.stakind3 THEN array_to_string(stavalues3, '||')
            WHEN 10002 = s.stakind4 THEN array_to_string(stavalues4, '||')
            ELSE NULL::varchar
        END AS pred_ts
   FROM pg_statistic s
   JOIN pg_class c ON c.oid = s.starelid
   JOIN pg_namespace ns ON c.relnamespace = ns.oid
   JOIN pg_attribute a ON c.oid = a.attrelid AND a.attnum = s.staattnum)
SELECT schema_name, table_name, col_num, col_name,
       pred_ts NOT LIKE '2000-01-01%' AS is_predicate,
       CASE WHEN pred_ts NOT LIKE '2000-01-01%' THEN (split_part(pred_ts,
 '||',1))::timestamp ELSE NULL::timestamp END as first_predicate_use,
       CASE WHEN pred_ts NOT LIKE '%||2000-01-01%' THEN (split_part(pred_ts,
 '||',2))::timestamp ELSE NULL::timestamp END as last_analyze
FROM predicate_column_info;
```

Presumi di eseguire la seguente query sulla tabella LISTING. Tieni presente che LISTID, LISTTIME ed EVENTID si utilizzano in un join, una condizione del filtro e una clausola di raggruppamento.

```
select s.buyerid,l.eventid, sum(l.totalprice)
from listing l
join sales s on l.listid = s.listid
where l.listtime > '2008-12-01'
group by l.eventid, s.buyerid;
```

Quando esegui una query sulla vista PREDICATE_COLUMNS, come illustrato nell'esempio seguente, puoi notare che LISTID, EVENTID e LISTTIME sono contrassegnati come colonne del predicato.

```
select * from predicate_columns
where table_name = 'listing';
```

```
schema_name | table_name | col_num | col_name       | is_predicate |
 first_predicate_use | last_analyze
------------+------------+---------+----------------+--------------
+---------------------+---------------------
```

```
public      | listing    |       1 | listid         | true         | 2017-05-05
 19:27:59 | 2017-05-03 18:27:41
public      | listing    |       2 | sellerid       | false        |
  | 2017-05-03 18:27:41
public      | listing    |       3 | eventid        | true         | 2017-05-16
 20:54:32 | 2017-05-03 18:27:41
public      | listing    |       4 | dateid         | false        |
  | 2017-05-03 18:27:41
public      | listing    |       5 | numtickets     | false        |
  | 2017-05-03 18:27:41
public      | listing    |       6 | priceperticket | false        |
  | 2017-05-03 18:27:41
public      | listing    |       7 | totalprice     | false        |
  | 2017-05-03 18:27:41
public      | listing    |       8 | listtime       | true         | 2017-05-16
 20:54:32 | 2017-05-03 18:27:41
```

Mantenere aggiornate le statistiche migliora le prestazioni delle query perché consente al pianificatore di query di scegliere i piani ottimali. Amazon Redshift aggiorna le statistiche automaticamente in background e consente di eseguire esplicitamente il comando ANALYZE. Se scegli di eseguire esplicitamente ANALYZE, procedi come descritto di seguito:

- Esegui il comando ANALYZE prima di eseguire le query.
- Esegui il comando ANALYZE sul database regolarmente alla fine di ogni normale ciclo di caricamento o aggiornamento.
- Esegui il comando ANALYZE su qualsiasi nuova tabella che crei e su qualsiasi tabella o colonna esistente soggetta a modifiche significative.
- Prendi in considerazione l'esecuzione di operazioni di ANALYZE su pianificazioni diverse per diversi tipi di tabelle e colonne, a seconda del loro utilizzo nelle query e della loro propensione al cambiamento.
- Per risparmiare tempo e risorse cluster, utilizza la clausola PREDICATE COLUMNS quando esegui ANALYZE.

Non è necessario eseguire esplicitamente il comando ANALYZE dopo aver ripristinato un'istantanea in un cluster fornito o in uno spazio dei nomi serverless, né dopo aver ripreso un cluster con provisioning sospeso. Amazon Redshift conserva le informazioni della tabella di sistema in questi casi, rendendo superflui i comandi ANALYZE manuali. Amazon Redshift continuerà a eseguire operazioni di analisi automatiche in base alle esigenze.

Un'operazione di analisi ignora le tabelle che contengono up-to-date statistiche. Se esegui ANALYZE nel flusso di lavoro ETL (Extract, Transform and Load, estrazione, trasformazione e caricamento), l'analisi automatica ignora le tabelle con statistiche aggiornate. Analogamente, un comando ANALYZE esplicito ignora le tabelle se l'analisi automatica ha aggiornato le statistiche della tabella.

Cronologia del comando ANALYZE

È utile sapere quando è stato eseguito l'ultima volta il comando ANALYZE su una tabella o un database. Quando viene eseguito un comando ANALYZE, Amazon Redshift esegue più query simili alla seguente:

```
padb_fetch_sample: select * from table_name
```

Esegui una query su STL_ANALYZE per visualizzare la cronologia delle operazioni di analisi. Se Amazon Redshift analizza una tabella con l'analisi automatica, la colonna `is_background` sarà impostata su `t` (true). Altrimenti, è impostata su `f` (false). L'esempio seguente collega STV_TBL_PERM per mostrare il nome della tabella e i dettagli di runtime.

```
select distinct a.xid, trim(t.name) as name, a.status, a.rows, a.modified_rows,
 a.starttime, a.endtime
from stl_analyze a
join stv_tbl_perm t  on t.id=a.table_id
where name = 'users'
order by starttime;

xid    | name  | status          | rows  | modified_rows | starttime           |
 endtime
-------+-------+-----------------+-------+---------------+---------------------
+---------------------
  1582 | users | Full            | 49990 |         49990 | 2016-09-22 22:02:23 |
 2016-09-22 22:02:28
244287 | users | Full            | 24992 |         74988 | 2016-10-04 22:50:58 |
 2016-10-04 22:51:01
244712 | users | Full            | 49984 |         24992 | 2016-10-04 22:56:07 |
 2016-10-04 22:56:07
245071 | users | Skipped         | 49984 |             0 | 2016-10-04 22:58:17 |
 2016-10-04 22:58:17
245439 | users | Skipped         | 49984 |          1982 | 2016-10-04 23:00:13 |
 2016-10-04 23:00:13
(5 rows)
```

In alternativa, è possibile eseguire una query più complessa che restituisce tutte le istruzioni eseguite in ogni transazione completata che includeva un comando ANALYZE:

```
select xid, to_char(starttime, 'HH24:MM:SS.MS') as starttime,
datediff(sec,starttime,endtime ) as secs, substring(text, 1, 40)
from svl_statementtext
where sequence = 0
and xid in (select xid from svl_statementtext s where s.text like 'padb_fetch_sample
%' )
order by xid desc, starttime;

xid  |  starttime   | secs |                 substring
-----+--------------+------+-----------------------------------------
1338 | 12:04:28.511 |    4 | Analyze date
1338 | 12:04:28.511 |    1 | padb_fetch_sample: select count(*) from
1338 | 12:04:29.443 |    2 | padb_fetch_sample: select * from date
1338 | 12:04:31.456 |    1 | padb_fetch_sample: select * from date
1337 | 12:04:24.388 |    1 | padb_fetch_sample: select count(*) from
1337 | 12:04:24.388 |    4 | Analyze sales
1337 | 12:04:25.322 |    2 | padb_fetch_sample: select * from sales
1337 | 12:04:27.363 |    1 | padb_fetch_sample: select * from sales
...
```

Vacuum delle tabelle

Amazon Redshift può ordinare ed eseguire automaticamente un'operazione VACUUM DELETE su tabelle in background. Per ripulire le tabelle dopo un caricamento o una serie di aggiornamenti incrementali, è anche possibile eseguire il comando VACUUM, sia sull'intero database, sia su singole tabelle.

> **Note**
>
> Solo gli utenti con le autorizzazioni necessarie possono efficacemente eseguire l'operazione vacuum su una tabella. Se VACUUM viene eseguito senza le necessarie autorizzazioni di tabella, l'operazione viene completata ma non ha alcun effetto. Per l'elenco delle autorizzazioni di tabella valide per eseguire efficacemente VACUUM, consulta VACUUM. Per questo motivo, raccomandiamo di eseguire il VACUUM sulle specifiche tabelle secondo necessità. Consigliamo questo approccio anche perché l'operazione VACUUM dell'intero database è potenzialmente un'operazione costosa.

Ordinamento automatico delle tabelle

Amazon Redshift ordina automaticamente i dati in background per mantenerli ordinati nella tabella rispetto alla chiave di ordinamento. Amazon Redshift tiene traccia delle query di scansione per determinare quali sezioni della tabella trarranno vantaggio dall'ordinamento.

In base al carico sul sistema, Amazon Redshift avvia automaticamente l'ordinamento. Questa operazione di ordinamento automatica riduce la necessità di eseguire il comando VACUUM per mantenere i dati ordinati secondo la chiave di ordinamento. Se sono necessari dati completamente ordinati in base alla chiave di ordinamento, ad esempio dopo un grande caricamento di dati, è comunque possibile eseguire manualmente il comando VACUUM. Per capire se la tabella trarrà benefici dall'esecuzione del comando VACUUM SORT, monitorare la colonna `vacuum_sort_benefit` in SVV_TABLE_INFO.

Amazon Redshift tiene traccia delle query di scansione che utilizzano la chiave di ordinamento su ogni tabella. Amazon Redshift stima la percentuale massima di miglioramento nella scansione e nel filtraggio dei dati per ciascuna tabella (se la tabella è stata completamente ordinata). Tale stima è visibile nella colonna `vacuum_sort_benefit` in SVV_TABLE_INFO. È possibile usare questa colonna, insieme alla colonna `unsorted`, per determinare quali query possano beneficare dell'esecuzione manuale del comando VACUUM SORT su un tavolo La colonna `unsorted` riflette l'ordinamento fisico di una tabella. La colonna `vacuum_sort_benefit` specifica l'impatto dell'ordinamento di una tabella tramite l'esecuzione manuale del comando VACUUM SORT.

Considerare, ad esempio, lo schema seguente:

```
select "table", unsorted,vacuum_sort_benefit from svv_table_info order by 1;
```

```
 table | unsorted | vacuum_sort_benefit
-------+----------+---------------------
 sales |    85.71 |                5.00
 event |    45.24 |               67.00
```

Per la tabella "vendite", sebbene la tabella sia fisicamente non ordinata per circa l'86%, l'impatto sulle prestazioni della query dovuto al fatto che la tabella non sia ordinata per l'86% è solo del 5%. Ciò potrebbe essere dovuto al fatto le query accedono solo a una piccola parte della tabella, o che pochissime query hanno dovuto accedere alla tabella. Per la tabella "evento", la tabella è fisicamente non ordinata per circa il 45%. Ma l'impatto sulle prestazioni della query del 67% indica che le query accedono a una porzione maggiore della tabella o che il numero di query che accedono alla tabella è

elevato. La tabella "evento" può ottenere potenzialmente dei vantaggi dall'esecuzione del comando VACUUM SORT.

Eliminazione vacuum automatica

Quando si esegue un'eliminazione, le righe vengono contrassegnate per l'eliminazione, ma non rimosse. Amazon Redshift esegue automaticamente un'operazione VACUUM DELETE in background in base al numero di righe eliminate nelle tabelle del database. Amazon Redshift pianificherà l'esecuzione di VACUUM DELETE durante i periodi di carico ridotto e sospenderà l'operazione durante i periodi di alto carico.

Argomenti

Frequenza di VACUUM

È opportuno eseguire l'operazione vacuum quando necessario per mantenere prestazioni di query coerenti. Prendi in considerazione questi fattori quando determini la frequenza con cui eseguire il comando VACUUM:

- Esegui il comando VACUUM durante i periodi di tempo in cui prevedi un'attività minima sul cluster, ad esempio la sera o durante le finestre di amministrazione del database designate.
- Esegui i comandi VACUUM al di fuori delle finestre di manutenzione. Per ulteriori informazioni, consulta Pianificazione per le finestre di manutenzione.
- Un'ampia regione non ordinata comporta tempi di vacuum più lunghi. Se ritardi il vacuum, questo richiederà più tempo perché è necessario riorganizzare una quantità maggiore di dati.
- VACUUM è un'operazione di I/O intensiva, quindi maggiore è il tempo richiesto per il completamento del vacuum, maggiore sarà l'impatto che avrà sulle query e su altre operazioni di database simultanee in esecuzione sul cluster.
- VACUUM richiede più tempo per le tabelle che utilizzano l'ordinamento interleaved. Per valutare se è necessario riordinare le tabelle interlacciate, esegui una query sulla vista SVV_INTERLEAVED_COLUMNS.

Fase di ordinamento e fase di unione

Amazon Redshift esegue un'operazione vacuum in due fasi: innanzitutto, ordina le righe nella regione non ordinata, quindi, se necessario, unisce le righe appena ordinate alla fine della tabella alle righe esistenti. Durante il vacuum di una tabella di grandi dimensioni, l'operazione di vacuum procede in una serie di passaggi costituiti da ordinamenti incrementali seguiti da unioni. Se l'operazione non riesce o se Amazon Redshift passa offline durante l'operazione vacuum, la tabella o il database parzialmente sottoposti a vacuum saranno in uno stato coerente, ma sarà necessario riavviare manualmente l'operazione vacuum. Gli ordinamenti incrementali vengono persi, ma non è necessario eseguire nuovamente il vacuum delle righe unite di cui è stato eseguito il commit prima dell'errore. Se la regione non ordinata è grande, l'operazione potrebbe richiedere molto tempo. Per ulteriori informazioni sulle fasi di ordinamento e unione, consultare Gestione del volume delle righe unite.

Gli utenti possono accedere alle tabelle mentre ne viene eseguito il vacuum. È possibile eseguire query e scrivere operazioni mentre viene eseguito il vacuum di una tabella, ma quando il DML e il vacuum vengono eseguiti contemporaneamente, entrambe le operazioni potrebbero richiedere più tempo. Se si eseguono le istruzioni UPDATE e DELETE durante un vacuum, le prestazioni del sistema potrebbero essere ridotte. Le unioni incrementali bloccano temporaneamente le operazioni simultanee di UPDATE e DELETE e le operazioni di UPDATE e DELETE a loro volta bloccano temporaneamente le fasi di unione incrementale sulle tabelle interessate. Le operazioni di DDL, come ALTER TABLE, vengono bloccate fino al termine dell'operazione di vacuum con la tabella.

> **Note**
>
> Vari modificatori per VACUUM controllano il modo in cui funziona. È possibile utilizzarli per personalizzare il funzionamento del vacuum in base alle esigenze attuali. Ad esempio, l'utilizzo di VACUUM RECLUSTER riduce l'operazione di vuoto non eseguendo un'operazione di fusione completa. Per ulteriori informazioni, consulta VACUUM.

Soglia di vacuum

Per impostazione predefinita, il comando VACUUM ignora la fase di ordinamento per ogni tabella dove più del 95% delle righe della tabella sono già ordinate. Ignorare la fase di ordinamento può migliorare significativamente le prestazioni di VACUUM. Per modificare la soglia di ordinamento predefinita per una singola tabella, includi il nome della tabella e il parametro TO threshold PERCENT quando esegui il comando VACUUM.

Tipi di vacuum

Per ulteriori informazioni sui diversi tipi di vacuum, consultare VACUUM.

Gestione dei tempi di vacuum

A seconda della natura dei dati, consigliamo di seguire le pratiche in questa sezione per ridurre al minimo i tempi di vacuum.

Argomenti

Decisione sulla reindicizzazione

Spesso è possibile migliorare in modo significativo le prestazioni delle query utilizzando uno stile di ordinamento interleaved, ma le prestazioni nel tempo potrebbero peggiorare se la distribuzione dei valori nelle colonne delle chiavi di ordinamento cambia.

Quando inizialmente si carica una tabella interlacciata vuota utilizzando COPY o CREATE TABLE AS, Amazon Redshift crea automaticamente l'indice interlacciato. Se inizialmente carichi una tabella interleaved utilizzando INSERT, è necessario eseguire VACUUM REINDEX in seguito per inizializzare l'indice interleaved.

Nel corso del tempo, quando aggiungi righe con nuovi valori di chiave di ordinamento, le prestazioni potrebbero peggiorare se la distribuzione dei valori nelle colonne della chiave di ordinamento cambia. Se le nuove righe rientrano principalmente nell'intervallo dei valori delle chiavi di ordinamento esistenti, non è necessario reindicizzare. Esegui VACUUM SORT ONLY o VACUUM FULL per ripristinare l'ordinamento.

Il motore di query è in grado di utilizzare l'ordinamento per selezionare in modo efficiente i blocchi di dati che devono essere sottoposti a scansione per elaborare una query. Per un ordinamento interlacciato, Amazon Redshift analizza i valori della colonna chiave di ordinamento per determinare l'ordinamento ottimale. Se la distribuzione dei valori delle chiavi cambia o si differenzia, quando vengono aggiunte le righe, la strategia di ordinamento non sarà più ottimale e il vantaggio in termini

di prestazioni dell'ordinamento diminuirà. Per analizzare nuovamente la distribuzione delle chiavi di ordinamento è possibile eseguire un VACUUM REINDEX. L'operazione di reindicizzazione richiede molto tempo, quindi per decidere se una tabella trarrà vantaggio da una reindicizzazione, esegui una query sulla vista SVV_INTERLEAVED_COLUMNS.

Ad esempio, la seguente query mostra i dettagli per le tabelle che utilizzano chiavi di ordinamento interleaved.

```
select tbl as tbl_id, stv_tbl_perm.name as table_name,
col, interleaved_skew, last_reindex
from svv_interleaved_columns, stv_tbl_perm
where svv_interleaved_columns.tbl = stv_tbl_perm.id
and interleaved_skew is not null;

 tbl_id | table_name | col | interleaved_skew | last_reindex
--------+------------+-----+------------------+---------------------
 100048 | customer   |   0 |             3.65 | 2015-04-22 22:05:45
 100068 | lineorder  |   1 |             2.65 | 2015-04-22 22:05:45
 100072 | part       |   0 |             1.65 | 2015-04-22 22:05:45
 100077 | supplier   |   1 |             1.00 | 2015-04-22 22:05:45
(4 rows)
```

Il valore per `interleaved_skew` è un rapporto che indica la quantità di differenza. Il valore 1 indica nessuna differenza. Se la differenza è maggiore di 1,4, un VACUUM REINDEX di solito migliora le prestazioni a meno che la differenza non riguardi il set sottostante.

È possibile utilizzare il valore della data `last_reindex` per determinare quanto tempo è passato dall'ultima reindicizzazione.

Gestione delle dimensioni della regione non ordinata

La regione non ordinata aumenta quando carichi grandi quantità di nuovi dati in tabelle che contengono già dati o quando non esegui il vacuum delle tabelle come parte delle operazioni di manutenzione ordinaria. Per evitare operazioni di vacuum a lunga durata, utilizza le seguenti pratiche:

- Esegui le operazioni di vacuum su una pianificazione regolare.

  Se carichi le tabelle in piccoli incrementi (come gli aggiornamenti giornalieri che rappresentano una piccola percentuale del numero totale di righe nella tabella), eseguire regolarmente VACUUM contribuirà a garantire che le singole operazioni di vacuum siano rapide.

- Esegui prima il carico maggiore.

  Se è necessario caricare una nuova tabella con più operazioni di COPY, esegui prima il carico più grande. Quando esegui un caricamento iniziale in una tabella nuova o troncata, tutti i dati vengono caricati direttamente nella regione ordinata, quindi non è richiesto il vacuum.

- Tronca una tabella anziché eliminare tutte le righe.

  L'eliminazione di righe da una tabella non recupera lo spazio occupato dalle righe finché non esegui un'operazione di vacuum; tuttavia, il troncamento di una tabella svuota la tabella e recupera lo spazio su disco, quindi non è richiesto il vacuum. In alternativa, rilascia la tabella e creala nuovamente.

- Tronca o rilascia tabelle di test.

  Se stai caricando un numero limitato di righe in una tabella a scopo di test, non eliminare le righe al termine dell'operazione. Tronca, invece, la tabella e ricarica quelle righe come parte della successiva operazione di caricamento di produzione.

- Esegui una copia completa.

  Se una tabella che utilizza una tabella di chiavi di ordinamento composta presenta un'ampia regione non ordinata, una copia completa risulta molto più veloce di un vacuum. Una copia completa ricrea e ripopola una tabella tramite inserimento di massa, che ordina di nuovo automaticamente la tabella. Se una tabella ha una regione ampia e non ordinata, una copia completa è molto più veloce di un vacuum. Lo svantaggio è che non è possibile effettuare aggiornamenti simultanei durante un'operazione di copia completa, possibile invece durante un vacuum. Per ulteriori informazioni, consulta Best practice di Amazon Redshift per la progettazione di query.

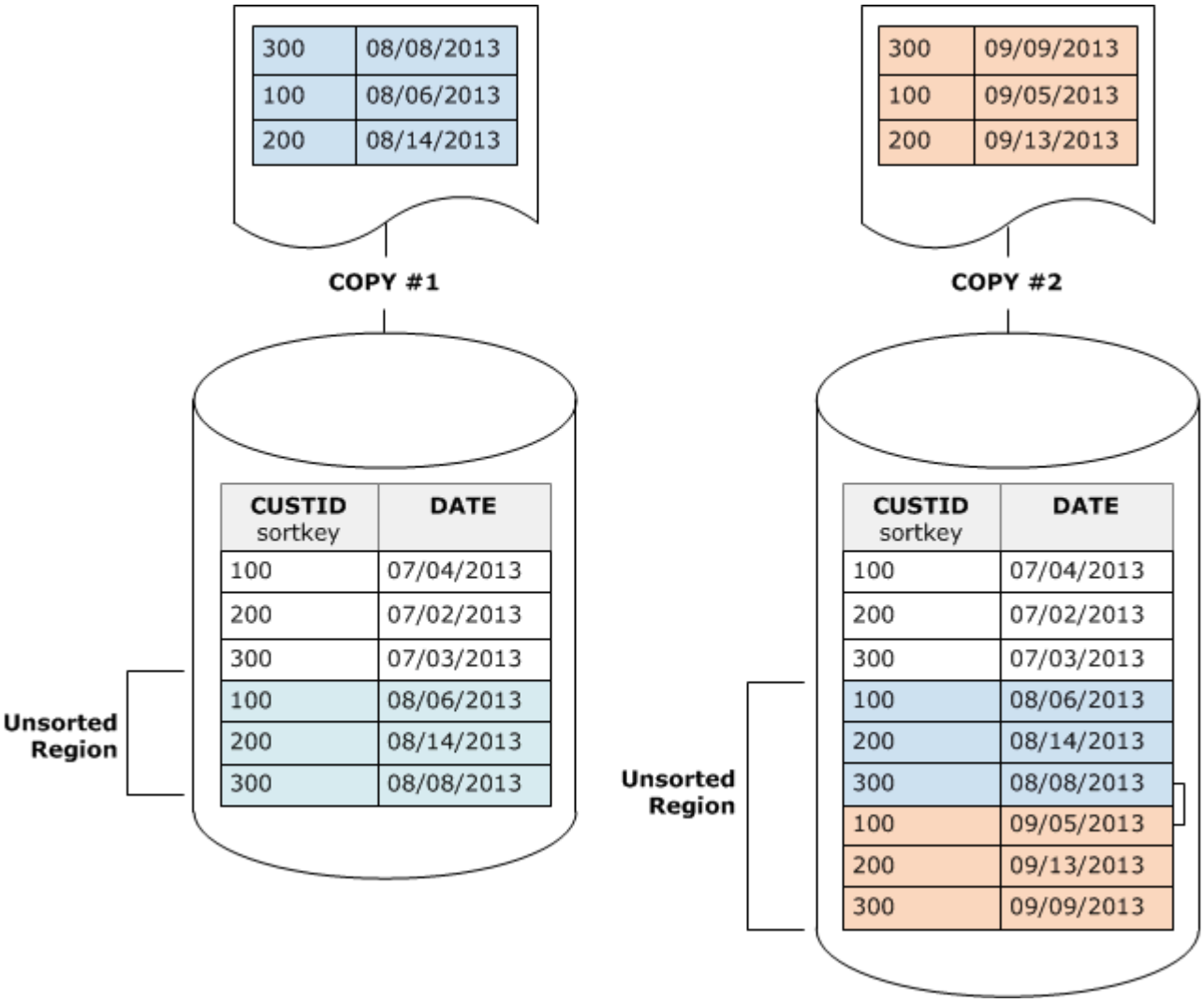
Gestione del volume delle righe unite

Se un'operazione di vacuum deve unire nuove righe nella regione ordinata di una tabella, il tempo richiesto per un vacuum aumenterà man mano che le dimensioni della tabella aumenteranno. È possibile migliorare le prestazioni di vacuum riducendo il numero di righe che devono essere unite.

Prima di un'operazione vacuum, una tabella è composta da una regione ordinata nella parte superiore della tabella, seguita da una regione non ordinata, che aumenta ogni volta che vengono aggiunte o aggiornate delle righe. Quando un set di righe viene aggiunto da un'operazione di COPY, il nuovo set di righe viene ordinato sulla chiave di ordinamento quando viene aggiunto alla regione

non ordinata nella parte inferiore della tabella. Le nuove righe sono ordinate all'interno dello stesso set, ma non all'interno della regione non ordinata.

Il diagramma seguente illustra la regione non ordinata dopo due successive operazioni di COPY, dove la chiave di ordinamento è CUSTID. Per semplicità, questo esempio mostra una chiave di ordinamento composta, ma gli stessi principi si applicano alle chiavi di ordinamento interleaved, tranne per il fatto che l'impatto della regione non ordinata è maggiore per le tabelle interleaved.



Un vacuum ripristina l'ordinamento della tabella in due fasi:

1. Ordina la regione non ordinata in una regione appena ordinata.

   La prima fase è relativamente economica, poiché viene riscritta solo la regione non ordinata. Se l'intervallo dei valori delle chiavi di ordinamento della regione appena ordinata è superiore all'intervallo esistente, solo le nuove righe devono essere riscritte e l'operazione vacuum è

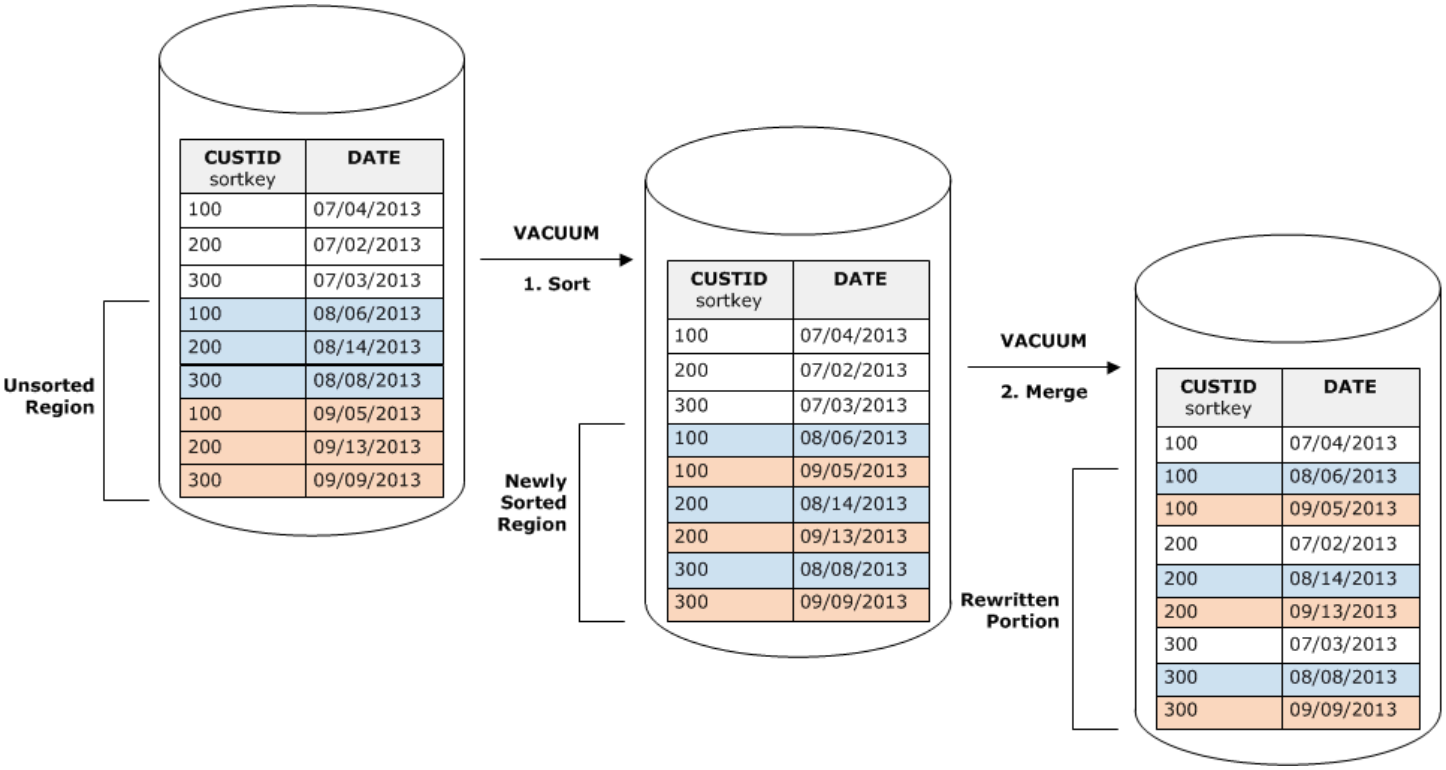
completa. Ad esempio, se la regione ordinata contiene valori di ID da 1 a 500 e le successive operazioni di COPY aggiungono valori chiave maggiori di 500, è necessario riscrivere solo la regione non ordinata.

2. Unisci la regione appena ordinata alla regione precedentemente ordinata.

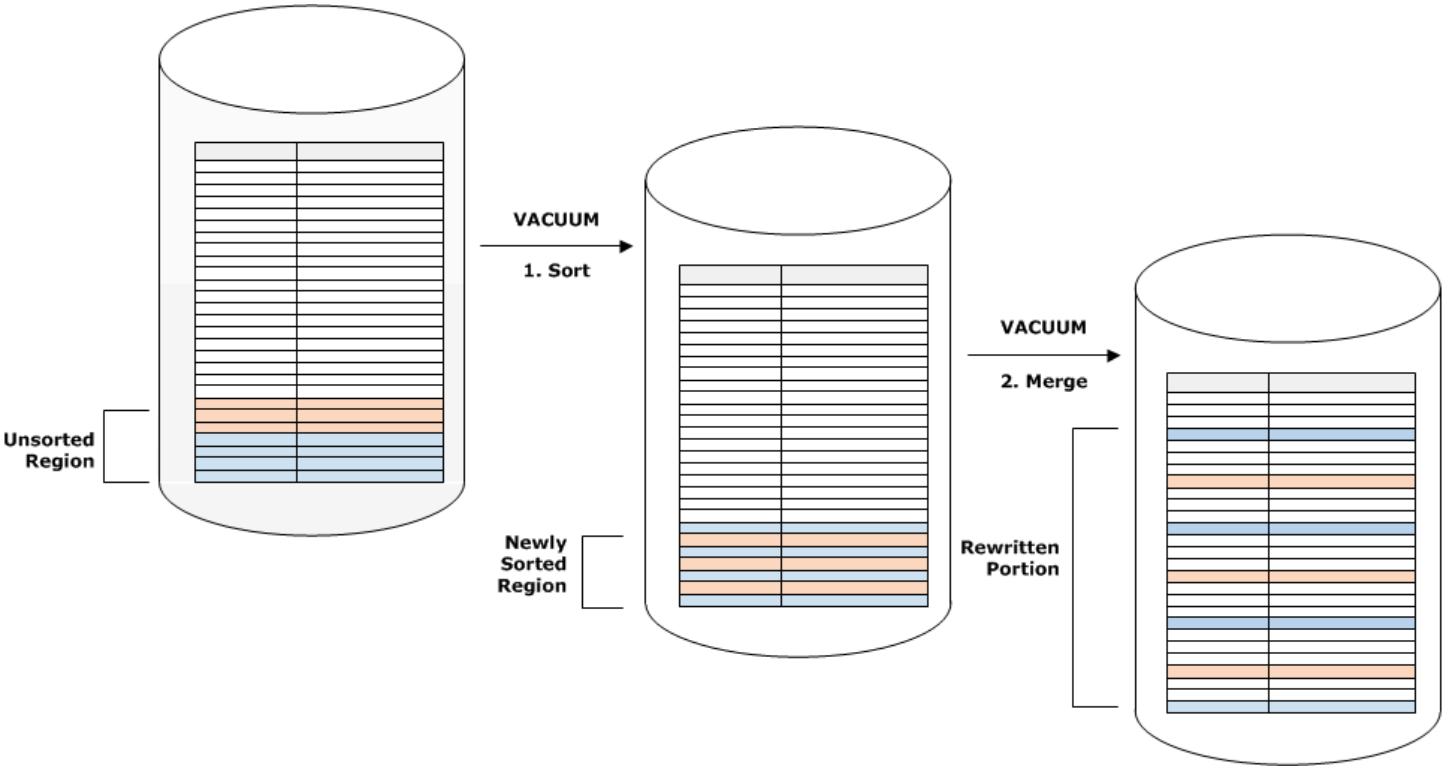
   Se le chiavi nella regione appena ordinata si sovrappongono alle chiavi nella regione ordinata, VACUUM deve unire le righe. Partendo dall'inizio della regione appena ordinata (con la chiave di ordinamento più bassa), il vacuum scrive le righe unite dalla regione ordinata in precedenza e la regione appena ordinata in una nuova serie di blocchi.

La misura in cui il nuovo intervallo di chiavi di ordinamento si sovrappone alle chiavi di ordinamento esistenti determina la misura in cui la regione precedentemente ordinata dovrà essere riscritta. Se le chiavi non ordinate sono distribuite nell'intero intervallo di ordinamento esistente, potrebbe essere necessario un vacuum per riscrivere le parti esistenti della tabella.

Il seguente diagramma mostra come un vacuum dovrebbe ordinare e unire righe che vengono aggiunte a una tabella in cui CUSTID è la chiave di ordinamento. Dato che ogni operazione di COPY aggiunge un nuovo set di righe con valori chiave che si sovrappongono alle chiavi esistenti, è necessario riscrivere quasi l'intera tabella. Il diagramma mostra una singola operazione di ordinamento e unione, ma in pratica, un'ampia operazione di vacuum consiste in una serie di fasi incrementali di ordinamento e unione.



Se l'intervallo di chiavi di ordinamento in un set di nuove righe si sovrappone all'intervallo di chiavi esistenti, il costo della fase di unione continua ad aumentare in proporzione alle dimensioni della tabella man mano che queste aumentano, mentre il costo della fase di ordinamento rimane proporzionale alla dimensione della regione non ordinata. In tal caso, il costo della fase di unione eclissa il costo della fase di ordinamento, come mostra il diagramma seguente.



Per determinare quale proporzione di una tabella è stata rimossa, esegui una query su SVV_VACUUM_SUMMARY al termine dell'operazione di vacuum. La seguente query mostra l'effetto di sei successive operazioni di vacuum man mano che CUSTSALES aumentava nel tempo.

```
select * from svv_vacuum_summary
where table_name = 'custsales';

 table_name | xid  | sort_      | merge_     | elapsed_   | row_  | sortedrow_ | block_
  | max_merge_
            |      | partitions | increments | time       | delta | delta      | delta
  | partitions
 -----------+------+------------+------------+------------+-------+------------
+---------+---------------
  custsales | 7072 |          3 |          2 |  143918314 |     0 |   88297472 |   1524
  |      47
  custsales | 7122 |          3 |          3 |  164157882 |     0 |   88297472 |    772
  |      47
  custsales | 7212 |          3 |          4 |  187433171 |     0 |   88297472 |    767
  |      47
  custsales | 7289 |          3 |          4 |  255482945 |     0 |   88297472 |    770
  |      47
  custsales | 7420 |          3 |          5 |  316583833 |     0 |   88297472 |    769
  |      47
```

```
 custsales | 9007 |          3 |          6 |  306685472 |     0 |   88297472 |    772
 |      47
(6 rows)
```

La colonna merge_increments fornisce un'indicazione della quantità di dati che è stata unita per ciascuna operazione di vacuum. Se il numero di incrementi di unione in operazioni vacuum consecutive aumenta in proporzione alla crescita delle dimensioni della tabella, ciò indica che ogni operazione vacuum sta unendo nuovamente un numero crescente di righe nella tabella, poiché le regioni esistenti e appena ordinate si sovrappongono.

Caricamento dei dati nell'ordine delle chiavi di ordinamento

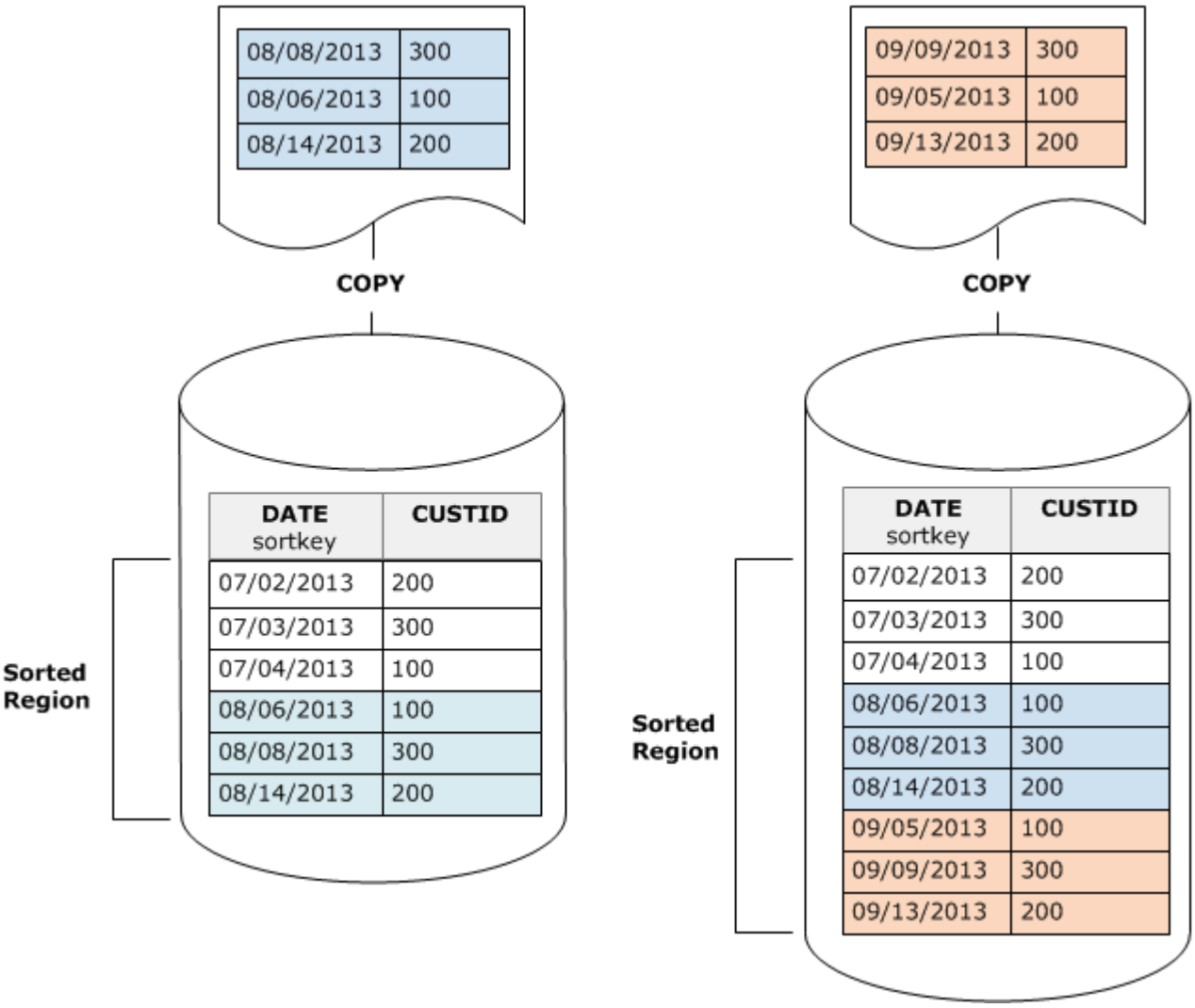
Se carichi i dati nell'ordine delle chiavi di ordinamento utilizzando un comando COPY, è possibile ridurre o anche eliminare la necessità di eseguire un'operazione vacuum.

COPY aggiunge automaticamente nuove righe alla regione ordinata della tabella quando sono vere tutte le seguenti condizioni:

- La tabella utilizza una chiave di ordinamento composta con una sola colonna di ordinamento.
- La colonna di ordinamento è NOT NULL.
- La tabella è ordinata al 100% o vuota.
- Tutte le nuove righe sono più alte nell'ordinamento rispetto alle righe esistenti, incluse le righe contrassegnate per l'eliminazione. In questo caso, Amazon Redshift utilizza i primi otto byte della chiave di ordinamento per determinare l'ordinamento.

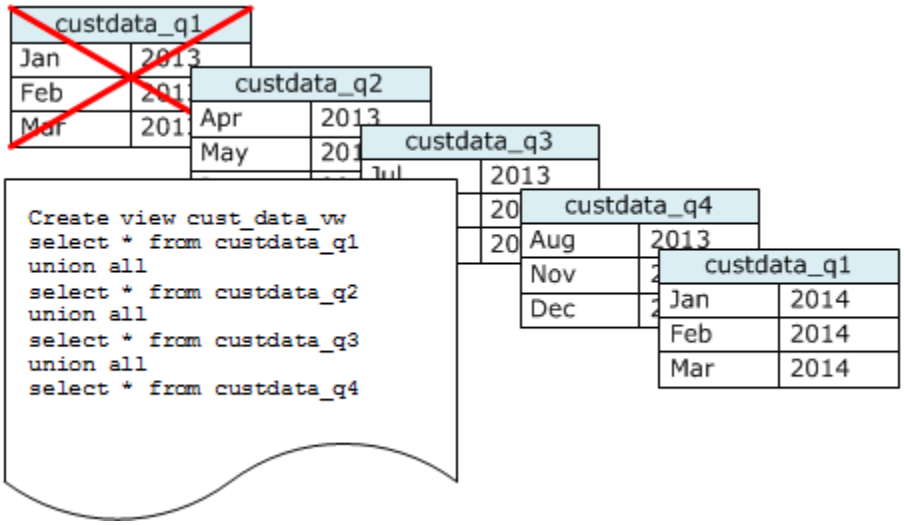
Ad esempio, supponiamo di disporre di una tabella che registra gli eventi dei clienti utilizzando un ID cliente e l'ora. Se ordini in base all'ID cliente, è probabile che l'intervallo di chiavi di ordinamento delle nuove righe aggiunte da carichi incrementali si sovrapponga all'intervallo esistente, come illustrato nell'esempio precedente, portando a una costosa operazione di vacuum.

Se imposti la chiave di ordinamento su una colonna timestamp, le nuove righe verranno aggiunte in ordine nella parte inferiore della tabella, come illustrato nel diagramma seguente, riducendo o addirittura eliminando la necessità di eseguire un'operazione vacuum.



Utilizzo di tabelle di serie temporali

Se conservi i dati per un periodo di tempo continuo, utilizza una serie di tabelle, come illustrato nel diagramma seguente.



Crea una nuova tabella ogni volta che aggiungi un set di dati, quindi elimina la tabella più vecchia della serie. Ottieni un doppio vantaggio:

- Eviti il costo aggiuntivo dell'eliminazione delle righe, poiché un'operazione di DROP TABLE è molto più efficiente di una di DELETE di massa.
- Se le tabelle sono ordinate per timestamp, non è necessario il vacuum. Se ogni tabella contiene dati per un mese, il vacuum dovrà al massimo riscrivere i dati di un mese, anche se le tabelle non sono ordinate per timestamp.

È possibile creare una vista UNION ALL per l'utilizzo segnalando query che nascondono il fatto che i dati sono archiviati in più tabelle. Se una query filtra la chiave di ordinamento, il pianificatore di query può ignorare in modo efficiente tutte le tabelle che non vengono utilizzate. Un UNION ALL può essere meno efficiente per altri tipi di query, pertanto è necessario valutare le prestazioni delle query nel contesto di tutte le query che utilizzano le tabelle.

Gestione delle operazioni di scrittura simultanee

Argomenti

- Isolamento serializzabile
- Operazioni di scrittura e lettura/scrittura
- Esempi di scrittura simultanea

Amazon Redshift consente la lettura delle tabelle mentre vengono modificate o caricate in modo incrementale.

In alcune tradizionali applicazioni di Data Warehousing e business intelligence, il database è disponibile agli utenti solo quando il caricamento notturno è completo. In questi casi, non sono consentiti aggiornamenti durante il normale orario di lavoro, quando vengono eseguite query analitiche e vengono generati report; tuttavia, un numero crescente di applicazioni rimane attivo per lunghi periodi del giorno o anche tutto il giorno, rendendo obsoleta la nozione di finestra di caricamento.

Amazon Redshift supporta questi tipi di applicazioni consentendo la lettura delle tabelle mentre vengono modificate o caricate in modo incrementale. Le query visualizzano semplicemente l'ultima versione di cui è stato eseguito il commit o la snapshot dei dati, invece di attendere che venga eseguito il commit della successiva versione. Se desideri che una determinata query attenda il commit da un'altra operazione di scrittura, è necessario pianificarla di conseguenza.

I seguenti argomenti descrivono alcuni concetti chiave e casi d'uso che riguardano transazioni, snapshot del database, aggiornamenti e comportamento simultaneo.

Isolamento serializzabile

Alcune applicazioni richiedono non solo query e caricamento simultanei, ma anche la possibilità di scrivere simultaneamente su più tabelle o sulla stessa tabella. In questo contesto, simultaneamente significa una sovrapposizione non pianificata per l'esecuzione simultanea. Due transazioni sono considerate simultanee se la seconda inizia prima del primo commit. Le operazioni simultanee possono provenire da sessioni diverse controllate dallo stesso utente o da utenti diversi.

> **Note**
>
> Amazon Redshift supporta un comportamento di commit automatico di default in cui viene eseguito singolarmente il commit di ogni comando SQL eseguito separatamente. Se si racchiude un set di comandi in un blocco di transazione (definito dalle istruzioni BEGIN e END), viene eseguito il commit del blocco come una transazione, quindi è possibile eseguirne il rollback se necessario. Eccezioni a questo comportamento sono i comandi TRUNCATE e VACUUM, che eseguono automaticamente il commit di tutte le modifiche in sospeso apportate nella transazione corrente.
> Alcuni client SQL eseguono automaticamente i comandi BEGIN e COMMIT, pertanto il client controlla se un gruppo di istruzioni viene eseguito come transazione o se ogni singola istruzione viene eseguita come transazione propria. Controlla la documentazione per l'interfaccia che stai utilizzando. Ad esempio, quando si utilizza il driver JDBC di Amazon Redshift, un'istruzione JDBC `PreparedStatement` con una stringa di query che contiene

più comandi SQL (separati da punto e virgola) esegue tutte le istruzioni come una singola transazione. Al contrario, se si utilizza SQL Workbench/J e si imposta AUTO COMMIT ON, quindi se si eseguono più istruzioni, ogni istruzione viene eseguita come transazione propria.

Le operazioni di scrittura simultanee sono supportate in Amazon Redshift in modo protettivo utilizzando i blocchi di scrittura sulle tabelle e il principio di isolamento serializzabile. L'isolamento serializzabile mantiene l'illusione che una transazione in esecuzione su una tabella sia l'unica transazione in esecuzione su quella tabella. Ad esempio, due transazioni in esecuzione simultanea, T1 e T2, devono produrre gli stessi risultati di almeno uno dei seguenti esempi:

- T1 e T2 vengono eseguite in serie in questo ordine
- T2 e T1 vengono eseguite in serie in questo ordine

Le transazioni simultanee sono invisibili l'una all'altra; non possono rilevare le reciproche modifiche. Ogni transazione simultanea creerà una snapshot del database all'inizio della transazione. Una snapshot del database viene creata all'interno di una transazione alla prima occorrenza della maggior parte delle istruzioni SELECT, dei comandi DML come COPY, DELETE, INSERT, UPDATE e TRUNCATE e dei seguenti comandi DDL:

- ALTER TABLE (per aggiungere o eliminare colonne)
- CREATE TABLE
- DROP TABLE
- TRUNCATE TABLE

Se qualsiasi esecuzione seriale delle transazioni simultanee produce gli stessi risultati della loro esecuzione simultanea, tali transazioni sono considerate "serializzabili" e possono essere eseguite in modo sicuro. Se nessuna esecuzione seriale di tali transazioni produce gli stessi risultati, la transazione che esegue un'istruzione che potrebbe interrompere la serializzabilità viene arrestata e ne viene eseguito il rollback.

Le tabelle del catalogo di sistema (PG) e altre tabelle di sistema Amazon Redshift (STL e STV) non sono bloccate in una transazione. Di conseguenza, le modifiche agli oggetti di database che derivano dalle operazioni di DDL e TRUNCATE sono visibili al commit in qualsiasi transazione simultanea.

Ad esempio, supponiamo che la tabella A esiste nel database quando iniziano due transazioni simultanee, T1 e T2. Supponiamo che T2 restituisca un elenco di tabelle selezionandole dalla

tabella catalogo PG_TABLES. Quindi T1 rilascia la tabella A ed esegue il commit, quindi T2 elenca nuovamente le tabelle. La tabella A non è più elencata. Se T2 prova a eseguire una query sulla tabella rilasciata, Amazon Redshift restituisce un errore di relazione inesistente. La query di catalogo che restituisce l'elenco di tabelle a T2 o verifica che la tabella A esista non è soggetta alle stesse regole di isolamento delle operazioni sulle tabelle utente.

Le transazioni per gli aggiornamenti a queste tabelle vengono eseguite in modalità di isolamento con lettura sottoposta al commit. Le tabelle del catalogo dei prefissi PG non supportano l'isolamento dello snapshot.

Isolamento serializzabile per tabelle di sistema e tabelle di catalogo

Uno snapshot del database viene creato in una transazione per qualsiasi query SELECT che fa riferimento a una tabella creata dall'utente o a una tabella di sistema Amazon Redshift (STL or STV). Le query SELECT che non fanno riferimento ad alcuna tabella non creano un nuovo snapshot del database delle transazioni. Le istruzioni INSERT, DELETE e UPDATE che operano esclusivamente sulle tabelle del catalogo di sistema (PG) non creano un nuovo snapshot del database delle transazioni.

Come correggere errori di isolamento serializzabile

ERROR:1023 DETAIL: violazione di isolamento serializzabile su una tabella in Redshift

Quando Amazon Redshift rileva un errore di isolamento serializzabile, compare un messaggio di errore simile al seguente.

```
ERROR:1023 DETAIL: Serializable isolation violation on table in Redshift
```

Per risolvere un errore di isolamento serializzabile, si possono tentare i seguenti metodi:

- Riprovare la transazione annullata.

  Amazon Redshift ha rilevato che un carico di lavoro simultaneo non è serializzabile. Suggerisce lacune nella logica dell'applicazione, che di solito possono essere risolte provando a eseguire di nuovo la transazione che ha riscontrato l'errore. Se il problema persiste, provare uno degli altri metodi.
- Spostare al di fuori della transazione tutte le operazioni che non devono essere nella stessa transazione atomica.

Questo metodo si applica quando singole operazioni all'interno di due transazioni fanno riferimento reciprocamente in modo tale da influire sul risultato dell'altra transazione. Ad esempio, ciascuna delle seguenti due sessioni avvia una transazione.

```
Session1_Redshift=# begin;
```

```
Session2_Redshift=# begin;
```

Il risultato di un'istruzione SELECT in una delle transazioni potrebbe essere compromesso da un'istruzione INSERT nell'altra. In altre parole, si presuppone che le seguenti istruzioni vengano eseguite in serie, in qualunque ordine. In ogni caso, il risultato è che una delle istruzioni SELECT restituisce una riga in più rispetto all'esecuzione simultanea delle transazioni. Non esiste un ordine in cui le operazioni in serie possono produrre lo stesso risultato dell'esecuzione simultanea. Di conseguenza, l'ultima operazione eseguita genera un errore di isolamento serializzabile.

```
Session1_Redshift=# select * from tab1;
Session1_Redshift=# insert into tab2 values (1);
```

```
Session2_Redshift=# insert into tab1 values (1);
Session2_Redshift=# select * from tab2;
```

In molti casi il risultato delle istruzioni SELECT non è importante. In altre parole, l'atomicità delle operazioni nelle transazioni non è importante. In questi casi, occorre spostare le istruzioni SELECT al di fuori delle transazioni, come illustrato nei seguenti esempi.

```
Session1_Redshift=# begin;
Session1_Redshift=# insert into tab1 values (1)
Session1_Redshift=# end;
Session1_Redshift=# select * from tab2;
```

```
Session2_Redshift # select * from tab1;
Session2_Redshift=# begin;
Session2_Redshift=# insert into tab2 values (1)
Session2_Redshift=# end;
```

In questi esempi non ci sono riferimenti reciproci nelle transazioni. Le due istruzioni INSERT non si compromettono reciprocamente. In questi esempi c'è almeno un ordine in cui le transazioni possono essere eseguite in serie producendo lo stesso risultato dell'esecuzione simultanea. Ciò vuol dire che le transazioni sono serializzabili.

- Forzare la serializzazione bloccando tutte le tabelle in ciascuna sessione.

  Il comando LOCK blocca le operazioni che possono generare errori di isolamento serializzabile. Quando utilizzi il comando LOCK, ricorda di fare quanto segue:

  - Bloccare tutte le tabelle interessate dalla transazione, incluse quelle interessate dalle istruzioni SELECT di sola lettura all'interno della transazione.
  - Bloccare le tabelle nello stesso ordine, indipendentemente da quello in cui vengono eseguite le operazioni.
  - Bloccare tutte le tabelle all'inizio delle transazione, prima di eseguire qualunque operazione.
- Utilizzo dell'isolamento degli snapshot per le transazioni simultanee

  Utilizza un comando ALTER DATABASE con isolamento degli snapshot. Per ulteriori informazioni sul parametro SNAPSHOT per ALTER DATABASE, consulta Parametri.

ERROR:1018 DETAIL: la relazione non esiste

Quando le operazioni simultanee di Amazon Redshift vengono eseguite in sessioni diverse, viene visualizzato un messaggio di errore simile al seguente.

```
ERROR: 1018 DETAIL: Relation does not exist.
```

Le transazioni in Amazon Redshift seguono l'isolamento degli snapshot. Dopo l'inizio di una transazione, Amazon Redshift acquisisce uno snapshot del database. Per l'intero ciclo di vita della transazione, la transazione opera sullo stato del database come indicato nello snapshot. Se la transazione legge da una tabella che non esiste nello snapshot, genera il messaggio di errore 1018 mostrato in precedenza. Anche quando un'altra transazione simultanea crea una tabella dopo che la transazione ha acquisito lo snapshot, la transazione non può leggere dalla tabella appena creata.

Per risolvere questo errore di isolamento della serializzazione, è possibile provare a spostare l'inizio della transazione in un punto in cui si sa che la tabella esiste.

Se la tabella viene creata da un'altra transazione, questo punto è almeno dopo il commit della transazione. Inoltre, assicurarsi che non sia stata eseguita alcuna transazione simultanea che potrebbe aver eliminato la tabella.

```
session1 = # BEGIN;
session1 = # DROP TABLE A;
session1 = # COMMIT;
```

```
session2 = # BEGIN;
```

```
session3 = # BEGIN;
session3 = # CREATE TABLE A (id INT);
session3 = # COMMIT;
```

```
session2 = # SELECT * FROM A;
```

Di conseguenza, l'ultima operazione eseguita come operazione di lettura da session2 genera un errore di isolamento serializzabile. Questo errore si verifica quando session2 esegue uno snapshot e la tabella è già stata eliminata da una sessione di commit 1. In altre parole, anche se una session3 simultanea ha creato la tabella, session2 non vede la tabella perché non è nello snapshot.

Per risolvere questo errore, è possibile ordinare nuovamente le sessioni come segue.

```
session1 = # BEGIN;
session1 = # DROP TABLE A;
session1 = # COMMIT;
```

```
session3 = # BEGIN;
session3 = # CREATE TABLE A (id INT);
session3 = # COMMIT;
```

```
session2 = # BEGIN;
session2 = # SELECT * FROM A;
```

Ora, quando session2 effettua il suo snapshot, per session3 è già stato eseguito il commit e la tabella è nel database. Session2 può leggere dalla tabella senza alcun errore.

Operazioni di scrittura e lettura/scrittura

È possibile gestire il comportamento specifico delle operazioni simultanee di scrittura decidendo quando e come eseguire diversi tipi di comandi. I seguenti comandi sono rilevanti a tal riguardo:

- Comandi COPY, che eseguono caricamenti (iniziali o incrementali)
- Comandi INSERT, che aggiungono una o più righe alla volta
- Comandi UPDATE, che modificano le righe esistenti
- Comandi DELETE, che rimuovono le righe esistenti

Le operazioni COPY e INSERT sono operazioni di scrittura pure, ma le operazioni DELETE e UPDATE sono operazioni di lettura/scrittura. Per poter essere eliminate o aggiornate, le righe devono prima essere lette. I risultati delle operazioni di scrittura simultanee dipendono dai comandi specifici che vengono eseguiti simultaneamente. Le operazioni di COPY e INSERT sulla stessa tabella vengono mantenute in uno stato di attesa fino al rilascio del blocco, quindi procedono normalmente.

Le operazioni di UPDATE e DELETE si comportano in modo diverso perché si basano su una tabella iniziale prima di eseguire qualsiasi scrittura. Dato che le transazioni simultanee sono invisibili l'una all'altra, sia UPDATE che DELETE devono leggere una snapshot dei dati dell'ultimo commit. Quando il primo UPDATE o DELETE rilascia il suo blocco, il secondo UPDATE o DELETE deve determinare se i dati con cui lavorerà sono potenzialmente obsoleti. Non saranno obsoleti, perché la seconda transazione non ottiene la sua snapshot di dati fino a quando la prima transazione non ha rilasciato il blocco.

Potenziale situazione di deadlock per le transazioni di scrittura simultanee

Ogni volta che le transazioni comportano aggiornamenti di più di una tabella, esiste sempre la possibilità di eseguire simultaneamente le transazioni in deadlock quando entrambe cercano di scrivere sullo stesso set di tabelle. Una transazione rilascia tutti i blocchi della tabella in una sola volta quando esegue il commit o il rollback; non rilascia i blocchi uno alla volta.

Ad esempio, supponiamo che le transazioni simultanee, T1 e T2 inizino circa nello stesso momento. Se T1 inizia a scrivere sulla tabella A e T2 inizia a scrivere sulla tabella B, entrambe le transazioni possono procedere senza conflitto; tuttavia, se T1 finisce di scrivere sulla tabella A e deve iniziare a scrivere sulla tabella B, non sarà in grado di procedere perché T2 mantiene ancora il blocco su B. Al contrario, se T2 finisce di scrivere sulla tabella B e deve iniziare a scrivere sulla tabella A, non sarà in grado di procedere perché T1 mantiene ancora il blocco su A. Poiché nessuna transazione può

rilasciare i blocchi finché tutte le operazioni di scrittura non vengono eseguite, l'altra transazione non può procedere.

Per evitare questo tipo di deadlock, è necessario pianificare attentamente le operazioni di scrittura simultanee. Ad esempio, è necessario aggiornare sempre le tabelle nello stesso ordine nelle transazioni e, se si specificano i blocchi, bloccare le tabelle nello stesso ordine prima di eseguire qualsiasi operazione di DML.

Esempi di scrittura simultanea

I seguenti esempi di pseudo codice dimostrano come, quando vengono eseguite simultaneamente, le transazioni procedono o attendono.

Operazioni simultanee di COPY nella stessa tabella

La transazione 1 copia le righe nella tabella LISTING:

```
begin;
copy listing from ...;
end;
```

La transazione 2 inizia simultaneamente in una sessione separata e prova a copiare più righe nella tabella LISTING. La transazione 2 deve attendere che la transazione 1 rilasci il blocco di scrittura sulla tabella LISTING, quindi può procedere.

```
begin;
[waits]
copy listing from ;
end;
```

Lo stesso comportamento si verificherebbe se una o entrambe le transazioni contenessero un comando INSERT anziché un comando COPY.

Operazioni simultanee di DELETE dalla stessa tabella

La transazione 1 elimina le righe da una tabella:

```
begin;
delete from listing where ...;
```

```
end;
```

La transazione 2 inizia simultaneamente e prova a eliminare le righe dalla stessa tabella. Avrà esito positivo perché attende il completamento della transazione 1 prima di provare eliminare le righe.

```
begin
[waits]
delete from listing where ;
end;
```

Lo stesso comportamento si verificherebbe se una o entrambe le transazioni contenessero un comando UPDATE nella stessa tabella anziché un comando DELETE.

Transazioni simultanee con una combinazione di operazioni di lettura e scrittura

In questo esempio, la transazione 1 elimina le righe dalla tabella USERS, ricarica la tabella, esegue una query su COUNT (*) e quindi ANALYZE prima di eseguire il commit:

```
begin;
delete one row from USERS table;
copy ;
select count(*) from users;
analyze ;
end;
```

Allo stesso tempo, inizia la transazione 2. Questa transazione prova a copiare righe aggiuntive nella tabella USERS, ad analizzare la tabella e quindi a eseguire la stessa query su COUNT (*) come la prima transazione:

```
begin;
[waits]
copy users from ...;
select count(*) from users;
analyze;
end;
```

La seconda transazione avrà esito positivo perché deve attendere il completamento della prima. La sua query su COUNT restituirà il conteggio in base al caricamento che ha completato.

Tutorial: Caricamento dei dati da Amazon S3

Questo tutorial guida attraverso l'intero processo di caricamento di dati nelle tabelle di database Amazon Redshift a partire dai file di dati in un bucket Amazon S3.

In questo tutorial, esegui quanto indicato di seguito:

- Scarichi file di dati che utilizzano formati CSV, delimitati da caratteri e a larghezza fissa.
- Creare un bucket Amazon S3 e quindi caricare i file di dati nel bucket.
- Avviare un cluster Amazon Redshift e creare le tabelle di database.
- Utilizzare i comandi COPY per caricare le tabelle dai file di dati su Amazon S3.
- Risolvere gli errori di caricamento e modificare i comandi COPY per correggere gli errori.

Tempo previsto: 60 minuti

Costo previsto: 1,00 USD/ora per il cluster

Prerequisiti

Sono necessari i seguenti prerequisiti:

- Un AWS account per avviare un cluster Amazon Redshift e creare un bucket in Amazon S3.
- AWS Le tue credenziali (ruolo IAM) per caricare i dati di test da Amazon S3. Se hai bisogno di un nuovo ruolo IAM, vai a Creazione di ruoli IAM.
- Un client SQL, ad esempio l'editor di query della console Amazon Redshift.

Questo tutorial è stato concepito in modo da essere svolto indipendentemente dagli altri. Oltre a questo tutorial, consigliamo di seguire i tutorial seguenti per avere una migliore comprensione del modo in cui progettare e utilizzare database Amazon Redshift.

- Il manuale Guida alle operazioni di base di Amazon Redshift guiderà attraverso il processo di creazione di un cluster Amazon Redshift e nel caricamento di dati di esempio.

Panoramica

È possibile aggiungere dati alle tabelle Amazon Redshift utilizzando un comando INSERT o COPY. Alla scala e alla velocità di un data warehouse Amazon Redshift, il comando COPY risulta molto più veloce ed efficace dei comandi INSERT.

Il comando COPY utilizza l'architettura MPP (Massively Parallel Processing) di Amazon Redshift per leggere e caricare dati in parallelo da più origini dati. È possibile caricare i file di dati su Amazon S3, Amazon EMR o qualsiasi host remoto accessibile mediante una connessione Secure Shell (SSH). In alternativa, è possibile caricare direttamente da una tabella Amazon DynamoDB.

In questo tutorial, sarà utilizzato il comando COPY per caricare dati da Amazon S3. Molti dei principi presentati qui sono validi anche per il caricamento da altre origini dati.

Per ulteriori informazioni sull'utilizzo del comando COPY, consultare le seguenti risorse:

- Best practice di Amazon Redshift per il caricamento di dati
- Caricamento di dati da Amazon EMR
- Caricamento di dati da host remoti
- Caricamento di dati da una tabella Amazon DynamoDB

Fasi

- Fase 1: creazione di un cluster
- Fase 2: download dei file di dati
- Fase 3: Caricamento dei file in un bucket Amazon S3
- Fase 4: creazione delle tabelle di esempio
- Fase 5: esecuzione dei comandi COPY
- Fase 6: vacuum e analisi del database
- Fase 7: elimina le risorse

Fase 1: creazione di un cluster

Se disponi già di un cluster che intendi utilizzare, puoi ignorare questa fase.

Per gli esercizi in questo tutorial, si utilizza un cluster a quattro nodi.

Come creare un cluster

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.

   Dal menu di navigazione, scegli Pannello di controllo dei cluster con provisioning.

   > **Important**
   >
   > Verifica di disporre delle autorizzazioni necessarie per eseguire le operazioni relative al cluster. Per informazioni sulla concessione delle autorizzazioni necessarie, consulta Autorizzazione di Amazon Redshift all'accesso ai servizi. AWS

2. In alto a destra, scegli la AWS regione in cui desideri creare il cluster. Ai fini del presente tutorial, selezionare Stati Uniti occidentali (Oregon).
3. Dal menu di navigazione, scegliere Clusters (Cluster), quindi Create cluster (Crea cluster). Appare la pagina Create cluster (Crea cluster).
4. Alla pagina Creazione di un cluster inserire i parametri per il cluster. Scegliere i tuoi valori per i parametri, eccetto modificare i seguenti valori:

   - Scegliere **`dc2.large`** per il tipo di nodo.
   - Scegliere **`4`** per Numero di nodi.
   - Nella sezione Cluster permissions (Autorizzazioni cluster), scegli un ruolo IAM da Available IAM roles (Ruoli IAM disponibili). Questo ruolo dovrebbe essere quello che creato in precedenza e che ha accesso ad Amazon S3. Quindi scegliere Add IAM role (Aggiungi ruolo IAM) per aggiungerlo all'elenco degli Attached IAM roles (Ruoli IAM collegati) del cluster.
5. Scegli Create cluster (Crea cluster).

Seguire la procedura riportata in Guida alle operazioni di base di Amazon Redshift per connettersi al cluster da un client SQL e testare una connessione. Non è necessario mettere in pratica le altre fasi indicate nella Guida Rapida relativa alla creazione di tabelle, al caricamento di dati e all'esecuzione di query di esempio.

Approfondimenti

Fase 2: download dei file di dati

Fase 2: download dei file di dati

In questa fase, scarichi un set di file di dati di esempio sul tuo computer. Nella fase successiva, i file verranno caricati in un bucket Amazon S3.

Per scaricare i file di dati

1. Scarica il file compresso: LoadingDataSampleFiles.zip.
2. Estrarre i file in una cartella sul computer.
3. Verificare che la cartella contenga i file seguenti.

```
customer-fw-manifest
customer-fw.tbl-000
customer-fw.tbl-000.bak
customer-fw.tbl-001
customer-fw.tbl-002
customer-fw.tbl-003
customer-fw.tbl-004
customer-fw.tbl-005
customer-fw.tbl-006
customer-fw.tbl-007
customer-fw.tbl.log
dwdate-tab.tbl-000
dwdate-tab.tbl-001
dwdate-tab.tbl-002
dwdate-tab.tbl-003
dwdate-tab.tbl-004
dwdate-tab.tbl-005
dwdate-tab.tbl-006
dwdate-tab.tbl-007
part-csv.tbl-000
part-csv.tbl-001
part-csv.tbl-002
part-csv.tbl-003
part-csv.tbl-004
part-csv.tbl-005
part-csv.tbl-006
part-csv.tbl-007
```

Approfondimenti

Fase 3: Caricamento dei file in un bucket Amazon S3

Fase 3: Caricamento dei file in un bucket Amazon S3

In questa fase, viene creato un bucket Amazon S3 e quindi i file di dati vengono caricati nel bucket.

Come caricare i file in un bucket Amazon S3

1. Creare un bucket in Amazon S3.

   Per ulteriori informazioni sulla creazione di un bucket, consulta Creazione di un bucket nella Guida per l'utente di Amazon Simple Storage Service.

   a. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
   b. Seleziona Crea bucket.
   c. Scegli un Regione AWS.

      Creare il bucket nella stessa regione del cluster. Se il cluster è nella regione Stati Uniti occidentali (Oregon), scegliere Regione Stati Uniti occidentali (Oregon) (us-west-2).
   d. Immetti un nome per il bucket nella casella Nome bucket della finestra di dialogo Crea bucket.

      Il nome di bucket scelto deve essere univoco tra tutti i nomi di bucket esistenti in Amazon S3. Una delle possibilità per garantire l'univocità consiste nell'aggiungere il nome dell'organizzazione come prefisso ai nomi dei bucket. I nomi di bucket devono soddisfare determinati requisiti. Per ulteriori informazioni, consultare Restrizioni e limitazioni dei bucket nella Guida per l'utente di Amazon Simple Storage Service.
   e. Scegli le impostazioni predefinite suggerite per le restanti opzioni.
   f. Seleziona Crea bucket.

      Una volta che Amazon S3 ha completato la creazione del bucket, la console visualizza il bucket vuoto nel pannello Bucket.
2. Creare una cartella.

   a. Scelta del nome del bucket.

b. Scegli il pulsante Crea cartella.

c. Assegnare il nome **`load`** alla nuova cartella.

> **Note**
>
> Il bucket creato non è in una sandbox. In questo esercizio aggiungi oggetti a un vero bucket. Per il tempo in cui conservi gli oggetti nel bucket, ti viene addebitato un importo nominale. Per maggiori informazioni sui prezzi di Amazon S3, consultare Prezzi di Amazon S3.

3. Caricare i file di dati nel nuovo bucket Amazon S3.

a. Scegliere il nome della cartella di dati.

b. Nella procedura guidata Carica scegli Aggiungi file.

Segui le istruzioni della console Amazon S3 per caricare tutti i file scaricati ed estratti,

c. Scegli Carica.

Credenziali utente

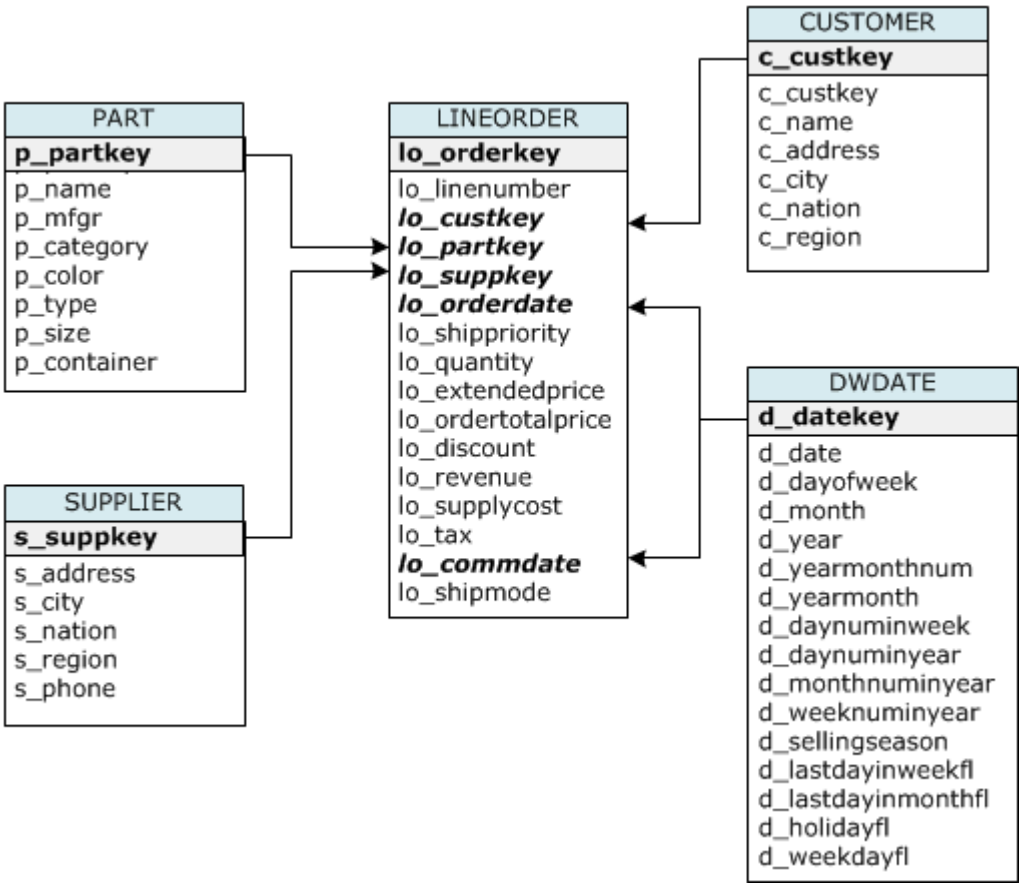
Il comando COPY di Amazon Redshift deve avere accesso in lettura agli oggetti file nel bucket Amazon S3. Se si utilizzano le stesse credenziali utente per creare il bucket Amazon S3 ed eseguire il comando COPY di Amazon Redshift, questo comando disporrà di tutte le autorizzazioni necessarie. Se si desidera utilizzare credenziali utente differenti, è possibile concedere l'accesso utilizzando i controlli accessi di Amazon S3. Il comando Amazon Redshift COPY richiede almeno due ListBucket GetObject autorizzazioni per accedere agli oggetti file nel bucket Amazon S3. Per ulteriori informazioni sul controllo degli accessi ad Amazon S3, consultare Gestione delle autorizzazioni di accesso alle risorse Amazon S3.

Approfondimenti

Fase 4: creazione delle tabelle di esempio

Fase 4: creazione delle tabelle di esempio

Per questo tutorial, utilizzerai un set di cinque tabelle basate sullo schema Star Schema Benchmark (SSB). Il diagramma seguente mostra il modello di dati SSB.



Le tabelle SSB potrebbero già essere presenti nel database corrente. In tal caso eliminale dal database prima di crearle con i comandi CREATE TABLE nella fase successiva. Le tabelle utilizzate in questo tutorial potrebbero avere attributi differenti rispetto alla tabelle esistenti.

Per creare le tabelle di esempio

1. Per eliminare le tabelle SSB, eseguire i seguenti comandi nel client SQL.

```
drop table part cascade;
drop table supplier;
drop table customer;
drop table dwdate;
drop table lineorder;
```

2. Eseguire i seguenti comandi CREATE TABLE nel client SQL.

```
CREATE TABLE part
(
  p_partkey     INTEGER NOT NULL,
  p_name        VARCHAR(22) NOT NULL,
```

```
  p_mfgr        VARCHAR(6),
  p_category    VARCHAR(7) NOT NULL,
  p_brand1      VARCHAR(9) NOT NULL,
  p_color       VARCHAR(11) NOT NULL,
  p_type        VARCHAR(25) NOT NULL,
  p_size        INTEGER NOT NULL,
  p_container   VARCHAR(10) NOT NULL
);

CREATE TABLE supplier
(
  s_suppkey   INTEGER NOT NULL,
  s_name      VARCHAR(25) NOT NULL,
  s_address   VARCHAR(25) NOT NULL,
  s_city      VARCHAR(10) NOT NULL,
  s_nation    VARCHAR(15) NOT NULL,
  s_region    VARCHAR(12) NOT NULL,
  s_phone     VARCHAR(15) NOT NULL
);

CREATE TABLE customer
(
  c_custkey      INTEGER NOT NULL,
  c_name         VARCHAR(25) NOT NULL,
  c_address      VARCHAR(25) NOT NULL,
  c_city         VARCHAR(10) NOT NULL,
  c_nation       VARCHAR(15) NOT NULL,
  c_region       VARCHAR(12) NOT NULL,
  c_phone        VARCHAR(15) NOT NULL,
  c_mktsegment   VARCHAR(10) NOT NULL
);

CREATE TABLE dwdate
(
  d_datekey            INTEGER NOT NULL,
  d_date               VARCHAR(19) NOT NULL,
  d_dayofweek          VARCHAR(10) NOT NULL,
  d_month              VARCHAR(10) NOT NULL,
  d_year               INTEGER NOT NULL,
  d_yearmonthnum       INTEGER NOT NULL,
  d_yearmonth          VARCHAR(8) NOT NULL,
  d_daynuminweek       INTEGER NOT NULL,
  d_daynuminmonth      INTEGER NOT NULL,
  d_daynuminyear       INTEGER NOT NULL,
```

```
  d_monthnuminyear     INTEGER NOT NULL,
  d_weeknuminyear      INTEGER NOT NULL,
  d_sellingseason      VARCHAR(13) NOT NULL,
  d_lastdayinweekfl    VARCHAR(1) NOT NULL,
  d_lastdayinmonthfl   VARCHAR(1) NOT NULL,
  d_holidayfl          VARCHAR(1) NOT NULL,
  d_weekdayfl          VARCHAR(1) NOT NULL
);
CREATE TABLE lineorder
(
  lo_orderkey          INTEGER NOT NULL,
  lo_linenumber        INTEGER NOT NULL,
  lo_custkey           INTEGER NOT NULL,
  lo_partkey           INTEGER NOT NULL,
  lo_suppkey           INTEGER NOT NULL,
  lo_orderdate         INTEGER NOT NULL,
  lo_orderpriority     VARCHAR(15) NOT NULL,
  lo_shippriority      VARCHAR(1) NOT NULL,
  lo_quantity          INTEGER NOT NULL,
  lo_extendedprice     INTEGER NOT NULL,
  lo_ordertotalprice   INTEGER NOT NULL,
  lo_discount          INTEGER NOT NULL,
  lo_revenue           INTEGER NOT NULL,
  lo_supplycost        INTEGER NOT NULL,
  lo_tax               INTEGER NOT NULL,
  lo_commitdate        INTEGER NOT NULL,
  lo_shipmode          VARCHAR(10) NOT NULL
);
```

Approfondimenti

[Fase 5: esecuzione dei comandi COPY](#)

Fase 5: esecuzione dei comandi COPY

In questa fase, esegui i comandi COPY per caricare ognuna della tabelle nello schema SSB. Gli esempi relativi al comando COPY illustrano il caricamento da differenti formati di file mediante varie opzioni del comando COPY nonché la risoluzione degli errori di caricamento.

Argomenti

- [Sintassi del comando COPY](#)

- Caricamento delle tabelle SSB

Sintassi del comando COPY

La sintassi di base del comando COPY è descritta di seguito.

```
COPY table_name [ column_list ] FROM data_source CREDENTIALS access_credentials
 [options]
```

Per eseguire un comando COPY, fornisci i valori seguenti.

Nome tabella

La tabella di destinazione per il comando COPY. La tabella deve esistere già nel database. La tabella può essere temporanea o persistente. Il comando COPY aggiunge i nuovi dati di input a tutte le righe esistenti nella tabella.

Elenco di colonne

Per impostazione predefinita, COPY carica i campi dai dati di origine nelle colonne della tabella nell'ordine. Puoi eventualmente specificare un elenco di colonne, ovvero un elenco di nomi di colonna separati da virgole, per associare i campi dei dati a colonne specifiche. In questo tutorial, non utilizzi elenchi di colonne. Per ulteriori informazioni, consultare Column List nel riferimento del comando COPY.

Origine dati

È possibile utilizzare il comando COPY per caricare dati da un bucket Amazon S3, un cluster Amazon EMR, un host remoto mediante una connessione SSH oppure da una tabella Amazon DynamoDB. Per questo tutorial, i file di dati saranno caricati in un bucket Amazon S3. Durante il caricamento da Amazon S3 è necessario fornire il nome del bucket e la posizione dei file di dati. Per farlo, indicare un percorso dell'oggetto per i file di dati o la posizione di un file manifest che elenchi esplicitamente ogni file di dati e la sua posizione.

- Prefisso di chiave

  Un oggetto archiviato in Amazon S3 è identificato in modo univoco da una chiave oggetto che include il nome del bucket, gli eventuali nomi delle cartelle e il nome dell'oggetto. Un prefisso della chiave fa riferimento a un insieme di oggetti con lo stesso prefisso. Il percorso oggetto è un prefisso

di chiave che il comando COPY utilizza per caricare tutti gli oggetti che condividono quel prefisso. Ad esempio, il prefisso di chiave `custdata.txt` può fare riferimento a un singolo file o a un insieme di file, ad esempio `custdata.txt.001`, `custdata.txt.002` e così di seguito.

- File manifest

  In alcuni casi, potrebbe essere necessario caricare file con prefissi diversi, ad esempio da più bucket o cartelle. In altri, potrebbe essere necessario escludere i file che condividono un prefisso. In questi casi, è possibile utilizzare un file manifest. Un file manifest elenca esplicitamente ogni file di caricamento e la relativa chiave oggetto univoca. Utilizzi un file manifest per caricare la tabella PART più avanti in questo tutorial.

Credenziali

Per accedere alle AWS risorse che contengono i dati da caricare, devi fornire le credenziali di AWS accesso a un utente con privilegi sufficienti. Queste credenziali includono un ruolo IAM (ARN) Amazon Resource Name (ARN). Per caricare dati da Amazon S3, le credenziali devono includere ListBucket e autorizzazioni. GetObject Ulteriori credenziali sono necessarie se i dati sono crittografati. Per ulteriori informazioni, consultare Parametri di autorizzazione nel riferimento del comando COPY. Per ulteriori informazioni sulla gestione degli accessi, consultare Gestione delle autorizzazioni di accesso alle risorse Amazon S3.

Opzioni

Puoi specificare vari parametri con il comando COPY per definire formati di file, gestire formati di dati, gestire errori e controllare altre funzionalità. In questo tutorial, utilizzi le seguenti opzioni e funzionalità del comando COPY:

- Prefisso di chiave

  Per informazioni su come caricare da più file specificando un prefisso chiave, consultare Caricamento della tabella PART mediante NULL AS.

- Formato CSV

  Per informazioni su come caricare i dati in formato CSV, consultare Caricamento della tabella PART mediante NULL AS.

- NULL AS

  Per informazioni su come caricare PART utilizzando l'opzione NULL AS, consultare Caricamento della tabella PART mediante NULL AS.

- Formato delimitato da carattere

  Per informazioni su come utilizzare l'opzione DELIMITER, consultare Caricamento della tabella SUPPLIER mediante REGION.

- REGION

  Per informazioni su come utilizzare l'opzione REGION, consultare Caricamento della tabella SUPPLIER mediante REGION.

- Larghezza di formato fisso

  Per informazioni su come caricare la tabella CUSTOMER dai dati a larghezza fissa, consultare Caricamento della tabella CUSTOMER mediante MANIFEST.

- MAXERROR

  Per informazioni su come utilizzare l'opzione MAXERROR, consultare Caricamento della tabella CUSTOMER mediante MANIFEST.

- ACCEPTINVCHARS

  Per informazioni su come utilizzare l'opzione ACCEPTINVCHARS, consultare Caricamento della tabella CUSTOMER mediante MANIFEST.

- MANIFEST

  Per informazioni su come utilizzare l'opzione MANIFEST, consultare Caricamento della tabella CUSTOMER mediante MANIFEST.

- DATEFORMAT

  Per informazioni su come utilizzare l'opzione DATEFORMAT, consultare Caricamento della tabella DWDATE mediante DATEFORMAT.

- GZIP, LZOP e BZIP2

  Per informazioni su come comprimere i file, consultare Caricamento della tabella LINEORDER mediante molteplici file.

- COMPUPDATE

  Per informazioni su come utilizzare l'opzione COMPUPDATE, consultare Caricamento della tabella LINEORDER mediante molteplici file.

- Molteplici file

Per informazioni su come caricare più file, consultare Caricamento della tabella LINEORDER mediante molteplici file.

Caricamento delle tabelle SSB

Per caricare ognuna delle tabelle nello schema SSB, utilizzi i comandi COPY seguenti. Ogni comando comporta opzioni COPY e tecniche di risoluzione dei problemi differenti.

Per caricare le tabelle SSB:

1. Sostituisci il nome e le credenziali del bucket AWS
2. Caricamento della tabella PART mediante NULL AS
3. Caricamento della tabella SUPPLIER mediante REGION
4. Caricamento della tabella CUSTOMER mediante MANIFEST
5. Caricamento della tabella DWDATE mediante DATEFORMAT
6. Caricamento della tabella LINEORDER mediante molteplici file

Sostituisci il nome e le credenziali del bucket AWS

I comandi COPY in questo tutorial hanno il formato illustrato di seguito.

```
copy table from 's3://<your-bucket-name>/load/key_prefix'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
options;
```

Per ogni comando COPY:

1. Sostituire *<your-bucket-name>* con il nome di un bucket nella stessa regione del cluster.

   Questa fase presuppone che il bucket e il cluster siano nella stessa regione. In alternativa, è possibile specificare regione utilizzando l'opzione REGION con il comando COPY.
2. Sostituisci <aws-account-id>e <role-name>con il tuo ruolo Account AWS e quello di IAM. Il segmento della stringa delle credenziali racchiuso tra virgolette singole non deve contenere spazi o interruzioni di riga. Tieni presente che l'ARN potrebbe differire leggermente nel formato rispetto all'esempio. È consigliabile copiare l'ARN per il ruolo dalla console IAM, per assicurarsi che sia accurato, quando esegui i comandi COPY.

Caricamento della tabella PART mediante NULL AS

In questa fase, utilizzi le opzioni CSV e NULL AS per caricare la tabella PART.

Il comando COPY consente il caricamento di dati da più file in parallelo, che è un'operazione molto più rapida rispetto al caricamento da un singolo file. Per illustrare questo principio, i dati di ogni tabella in questo tutorial sono suddivisi in otto file, anche se i file sono molto piccoli. In una fase successiva, confronterai la differenza di tempo tra il caricamento da un singolo file e da più file. Per ulteriori informazioni, consulta Caricamento di file di dati.

Prefisso di chiave

Puoi eseguire i caricamenti da più file specificando un prefisso di chiave per il set di file oppure elencando esplicitamente i file in un file manifest. In questa fase, utilizzerai un prefisso della chiave. In una fase successiva, utilizzerai un file manifest. Il prefisso di chiave `'s3://mybucket/load/part-csv.tbl'` carica il seguente set di file nella cartella `load`.

```
part-csv.tbl-000
part-csv.tbl-001
part-csv.tbl-002
part-csv.tbl-003
part-csv.tbl-004
part-csv.tbl-005
part-csv.tbl-006
part-csv.tbl-007
```

Formato CSV

CSV, acronimo di Comma Separated Values (valori separati da virgole), è un formato comune per l'importazione e l'esportazione di dati di fogli di calcolo. CSV è più flessibile del formato delimitato da virgole in quanto ti consente di includere stringhe tra virgolette nei campi. Le virgolette di delimitazione predefinite per COPY dal formato CSV sono le virgolette doppie ("), ma è possibile specificare un altro tipo di virgolette utilizzando l'opzione QUOTE AS. Quando si utilizzano le virgolette di delimitazione nel campo, creare una sequenza di escape aggiungendo virgolette di delimitazione aggiuntive.

Il seguente estratto da un file di dati in formato CSV della tabella PART mostra delle stringhe racchiuse tra virgolette doppie (`"LARGE ANODIZED BRASS"`). Mostra anche una stringa racchiusa tra doppie virgolette all'interno di una stringa di citazion (`"MEDIUM ""BURNISHED"" TIN"`).

```
15,dark sky,MFGR#3,MFGR#47,MFGR#3438,indigo,"LARGE ANODIZED BRASS",45,LG CASE
```

```
22,floral beige,MFGR#4,MFGR#44,MFGR#4421,medium,"PROMO, POLISHED BRASS",19,LG DRUM
23,bisque slate,MFGR#4,MFGR#41,MFGR#4137,firebrick,"MEDIUM ""BURNISHED"" TIN",42,JUMBO
 JAR
```

I dati della tabella PART contengono caratteri che generano un errore nell'esecuzione del comando COPY. In questo esercizio, individuiamo e correggiamo tali errori.

Per caricare i dati in formato CSV, aggiungi `csv` al comando COPY. Esegui il comando seguente per caricare la tabella PART.

```
copy part from 's3://<your-bucket-name>/load/part-csv.tbl'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
csv;
```

Potrebbe essere visualizzato un messaggio di errore simile al seguente.

```
An error occurred when executing the SQL command:
copy part from 's3://mybucket/load/part-csv.tbl'
credentials' ...

ERROR: Load into table 'part' failed.  Check 'stl_load_errors' system table for
 details. [SQL State=XX000]

Execution time: 1.46s

1 statement(s) failed.
1 statement(s) failed.
```

Per ottenere maggiori informazioni sull'errore, esegui una query sulla tabella STL_LOAD_ERRORS. La query seguente utilizza la funzione SUBSTRING per ridurre le colonne e facilitare la lettura e utilizza LIMIT 10 per ridurre il numero di righe restituite. Puoi regolare i valori in `substring(filename,22,25)` allo scopo di includere la lunghezza del nome di bucket.

```
select query, substring(filename,22,25) as filename,line_number as line,
substring(colname,0,12) as column, type, position as pos, substring(raw_line,0,30) as
 line_text,
substring(raw_field_value,0,15) as field_text,
substring(err_reason,0,45) as reason
from stl_load_errors
order by query desc
limit 10;
```

```
 query  |    filename      | line |  column   |    type    | pos |
--------+------------------------+----------+------------+------------+-----+----
 333765 | part-csv.tbl-000 |    1 |           |            |   0 |

 line_text        | field_text |                    reason
------------------+------------+-----------------------------------------------
 15,NUL next,     |            | Missing newline: Unexpected character 0x2c f
```

NULL AS

I fila di dati `part-csv.tbl` utilizzano il carattere di terminazione NUL (`\x000` o `\x0`) per indicare valori NULL.

> **Note**
>
> Nonostante un'ortografia molto simile, NUL e NULL non sono identici. NUL è un carattere UTF-8 con punto di codice `x000`, spesso utilizzato per indicare la fine del record. NULL è un valore SQL che rappresenta l'assenza di dati.

Per impostazione predefinita, COPY tratta un carattere di terminazione NUL come carattere di fine record e termina il record, generando spesso risultati imprevisti o un errore. Non esiste un singolo metodo standard per indicare NULL in un dato testuale. Perciò l'opzione NULL AS del comando COPY ti consente di specificare quale carattere sostituire con NULL durante il caricamento della tabella. In questo esempio, vuoi che COPY consideri il carattere di terminazione NUL come valore NULL.

> **Note**
>
> La colonna della tabella che riceve il valore NULL deve essere configurata come nullable Ciò significa che non deve includere il vincolo NOT NULL nella specifica CREATE TABLE.

Per caricare la tabella PART utilizzando l'opzione NULL AS, esegui il comando COPY seguente.

```
copy part from 's3://<your-bucket-name>/load/part-csv.tbl'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
csv
null as '\000';
```

Per verificare che COPY ha caricato i valori NULL, esegui il comando seguente per selezionare solo le righe che contengono NULL.

```
select p_partkey, p_name, p_mfgr, p_category from part where p_mfgr is null;
```

```
 p_partkey |  p_name  | p_mfgr | p_category
-----------+----------+--------+------------
        15 | NUL next |        | MFGR#47
        81 | NUL next |        | MFGR#23
       133 | NUL next |        | MFGR#44
(2 rows)
```

Caricamento della tabella SUPPLIER mediante REGION

In questa fase, utilizzi le opzioni DELIMITER e REGION per caricare la tabella SUPPLIER.

> **Note**
>
> I file per il caricamento della tabella SUPPLIER sono forniti in un bucket AWS di esempio. Non devi caricare i file per questa fase.

Formato delimitato da carattere

I campi in un file delimitato da carattere sono separati da uno specifico carattere, ad esempio una barra verticale (|), una virgola (,) o una tabulazione (\t). I file delimitati da carattere possono utilizzare qualsiasi carattere ASCII, inclusi i caratteri ASCII non stampabili, come delimitatore. Per specificare il carattere delimitatore, utilizzi l'opzione DELIMITER. Il delimitatore predefinito è una barra verticale (|).

L'estratto seguente dei dati della tabella SUPPLIER utilizza il formato delimitato da una barra verticale.

```
1|1|257368|465569|41365|19950218|2-HIGH|0|17|2608718|9783671|4|2504369|92072|2|
19950331|TRUCK
1|2|257368|201928|8146|19950218|2-HIGH|0|36|6587676|9783671|9|5994785|109794|6|
19950416|MAIL
```

REGION

Quando possibile, dovresti collocare i dati di carico nella stessa AWS regione del cluster Amazon Redshift. Se tali dati e il cluster si trovano nella stessa regione, si riduce la latenza e si evitano i costi di trasferimento dei dati tra regioni. Per ulteriori informazioni, consulta Best practice di Amazon Redshift per il caricamento di dati

Se devi caricare dati da una AWS regione diversa, utilizza l'opzione REGION per specificare la AWS regione in cui si trovano i dati di caricamento. Se specifichi una regione, tutti i dati di caricamento, inclusi i file manifest, devono trovarsi in tale regione. Per ulteriori informazioni, consulta REGION.

Se il cluster si trova nella regione Stati Uniti orientali (Virginia settentrionale), eseguire il comando seguente per caricare la tabella SUPPLIER a partire dai dati delimitati da barra verticale in un bucket Amazon S3 che si trova nella regione Stati Uniti occidentali (Oregon). Per questo esempio, non modificare il nome del bucket.

```
copy supplier from 's3://awssampledbuswest2/ssbgz/supplier.tbl'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
delimiter '|'
gzip
region 'us-west-2';
```

Se il cluster non si trova nella regione Stati Uniti orientali (Virginia settentrionale), eseguire il comando seguente per caricare la tabella SUPPLIER a partire dai dati delimitati da barra verticale in un bucket Amazon S3 che si trova nella regione Stati Uniti orientali (Virginia settentrionale). Per questo esempio, non modificare il nome del bucket.

```
copy supplier from 's3://awssampledb/ssbgz/supplier.tbl'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
delimiter '|'
gzip
region 'us-east-1';
```

Caricamento della tabella CUSTOMER mediante MANIFEST

In questa fase, utilizzi le opzioni FIXEDWIDTH, MAXERROR, ACCEPTINVCHARS e MANIFEST per caricare la tabella CUSTOMER.

I dati di esempio per questo esercizio contengono caratteri che genereranno errori quando il comando COPY tenta di caricarli. Utilizzi l'opzione MAXERRORS e la tabella di sistema STL_LOAD_ERRORS per risolvere gli errori di caricamento e quindi le opzioni ACCEPTINVCHARS e MANIFEST per eliminare gli errori.

Formato a larghezza fissa

Il formato a larghezza fissa definisce ogni campo come numero di caratteri fisso e non separa i campi con un delimitatore. L'estratto seguente dei dati della tabella CUSTOMER utilizza il formato a larghezza fissa.

```
1   Customer#000000001   IVhzIApeRb            MOROCCO  0MOROCCO  AFRICA      25-705
2   Customer#000000002   XSTf4,NCwDVaWNe6tE    JORDAN   6JORDAN   MIDDLE EAST 23-453
3   Customer#000000003   MG9kdTD               ARGENTINA5ARGENTINAAMERICA     11-783
```

L'ordine delle coppie etichetta/larghezza deve corrispondere esattamente all'ordine delle colonne della tabella. Per ulteriori informazioni, consulta FIXEDWIDTH.

La stringa di specifica a larghezza fissa per i dati della tabella CUSTOMER è la seguente.

```
fixedwidth 'c_custkey:10, c_name:25, c_address:25, c_city:10, c_nation:15,
c_region :12, c_phone:15,c_mktsegment:10'
```

Per caricare la tabella CUSTOMER a partire dai dati a larghezza, esegui il comando seguente.

```
copy customer
from 's3://<your-bucket-name>/load/customer-fw.tbl'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
fixedwidth 'c_custkey:10, c_name:25, c_address:25, c_city:10, c_nation:15,
 c_region :12, c_phone:15,c_mktsegment:10';
```

Dovrebbe essere visualizzato un messaggio di errore simile al seguente.

```
An error occurred when executing the SQL command:
copy customer
from 's3://mybucket/load/customer-fw.tbl'
credentials'...

ERROR: Load into table 'customer' failed.  Check 'stl_load_errors' system table for
 details. [SQL State=XX000]

Execution time: 2.95s

1 statement(s) failed.
```

MAXERROR

Per impostazione predefinita, al primo errore, il comando COPY non riesce e restituisce un messaggio di errore. Per risparmiare tempo durante il testing, puoi utilizzare l'opzione MAXERROR per indicare a COPY di ignorare un determinato numero di errori prima di avere esito negativo. Poiché ci aspettiamo degli errori la prima volta che testiamo il caricamento dei dati della tabella CUSTOMER, aggiungi `maxerror 10` al comando COPY.

Per eseguire il test utilizzando le opzioni FIXEDWIDTH e MAXERROR, esegui il comando seguente.

```
copy customer
from 's3://<your-bucket-name>/load/customer-fw.tbl'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
fixedwidth 'c_custkey:10, c_name:25, c_address:25, c_city:10, c_nation:15,
 c_region :12, c_phone:15,c_mktsegment:10'
maxerror 10;
```

Questa volta, anziché un messaggio di errore, viene restituito un messaggio simile al seguente.

```
Warnings:
Load into table 'customer' completed, 112497 record(s) loaded successfully.
Load into table 'customer' completed, 7 record(s) could not be loaded.  Check
 'stl_load_errors' system table for details.
```

L'avviso segnala che sono stati rilevati sette errori durante l'esecuzione di COPY. Per verificare gli errori, esegui una query sulla tabella STL_LOAD_ERRORS, come mostrato nell'esempio seguente.

```
select query, substring(filename,22,25) as filename,line_number as line,
substring(colname,0,12) as column, type, position as pos, substring(raw_line,0,30) as
 line_text,
substring(raw_field_value,0,15) as field_text,
substring(err_reason,0,45) as error_reason
from stl_load_errors
order by query desc, filename
limit 7;
```

I risultati della query STL_LOAD_ERRORS devono essere simili a quanto segue.

```
 query  |         filename          | line |  column   |    type    | pos |
 line_text           | field_text |              error_reason
--------+---------------------------+------+-----------
+------------+-----+-------------------------------+------------
+--------------------------------------------
```

```
 334489 | customer-fw.tbl.log       |    2 | c_custkey | int4       |  -1 | customer-
fw.tbl               | customer-f | Invalid digit, Value 'c', Pos 0, Type: Integ
 334489 | customer-fw.tbl.log       |    6 | c_custkey | int4       |  -1 | Complete
                  | Complete   | Invalid digit, Value 'C', Pos 0, Type: Integ
 334489 | customer-fw.tbl.log       |    3 | c_custkey | int4       |  -1 | #Total rows
                  | #Total row | Invalid digit, Value '#', Pos 0, Type: Integ
 334489 | customer-fw.tbl.log       |    5 | c_custkey | int4       |  -1 | #Status
                  | #Status    | Invalid digit, Value '#', Pos 0, Type: Integ
 334489 | customer-fw.tbl.log       |    1 | c_custkey | int4       |  -1 | #Load file
                  | #Load file | Invalid digit, Value '#', Pos 0, Type: Integ
 334489 | customer-fw.tbl000        |    1 | c_address | varchar    |  34 | 1
 Customer#000000001  | .Mayag.ezR | String contains invalid or unsupported UTF8
 334489 | customer-fw.tbl000        |    1 | c_address | varchar    |  34 | 1
 Customer#000000001  | .Mayag.ezR | String contains invalid or unsupported UTF8
(7 rows)
```

Esaminando i risultati, è possibile notare che vi sono due messaggi nella colonna `error_reasons`:

- `Invalid digit, Value '#', Pos 0, Type: Integ`

  Questi errori sono dovuti al file `customer-fw.tbl.log`. Il problema è che si tratta di un file di log e non di un file di dati, quindi non dovrebbe essere caricato. Puoi utilizzare un file manifest per evitare di caricare file non appropriati.

- `String contains invalid or unsupported UTF8`

  Il tipo di dati VARCHAR supporta caratteri UTF-8 di al massimo tre byte. Se i dati di caricamento contengono caratteri non supportati o non validi, puoi utilizzare l'opzione ACCEPTINVCHARS per sostituire ogni carattere non valido con un carattere alternativo specificato.

Un altro problema con il carico è più difficile da rilevare: il carico ha prodotto risultati imprevisti. Per analizzare questo problema, esegui una query sulla tabella CUSTOMER con il comando seguente.

```
select c_custkey, c_name, c_address
from customer
order by c_custkey
limit 10;
```

```
 c_custkey |          c_name           |         c_address
-----------+---------------------------+---------------------------
```

```
         2 | Customer#000000002        | XSTf4,NCwDVaWNe6tE
         2 | Customer#000000002        | XSTf4,NCwDVaWNe6tE
         3 | Customer#000000003        | MG9kdTD
         3 | Customer#000000003        | MG9kdTD
         4 | Customer#000000004        | XxVSJsL
         4 | Customer#000000004        | XxVSJsL
         5 | Customer#000000005        | KvpyuHCplrB84WgAi
         5 | Customer#000000005        | KvpyuHCplrB84WgAi
         6 | Customer#000000006        | sKZz0CsnMD7mp4Xd0YrBvx
         6 | Customer#000000006        | sKZz0CsnMD7mp4Xd0YrBvx
(10 rows)
```

Le righe devono essere univoche, ma in questo caso si hanno dei duplicati.

Un altro modo di rilevare risultati imprevisti è di verificare il numero di righe caricate. Nel nostro caso, avremmo dovuto avere 100000 righe caricate, ma il messaggio ha segnalato 112497 record. Le righe in eccesso sono state caricate in quanto il comando COPY ha caricato un file estraneo, `customer-fw.tbl0000.bak`.

In questo esercizio, utilizzi un file manifest per evitare di caricare i file non appropriati.

ACCEPTINVCHARS

Per impostazione predefinita, quando COPY incontra un carattere non supportato dal tipo di dati della colonna, ignora la riga e restituisce un errore. Per informazioni sui caratteri UTF-8 non validi, consultare Errori di caricamento di caratteri multibyte.

Potresti utilizzare l'opzione MAXERRORS per ignorare gli errori e continuare il caricamento, quindi eseguire una query su STL_LOAD_ERRORS per individuare i caratteri non validi e infine correggere i file di dati. Tuttavia, MAXERRORS è più appropriato per risolvere i problemi di caricamento e in genere non dovrebbe essere utilizzato in un ambiente di produzione.

L'opzione ACCEPTINVCHARS è generalmente una scelta più adatta per la gestione di caratteri non validi. ACCEPTINVCHARS indica a COPY di sostituire ogni carattere non valido con un carattere valido specificato e di proseguire l'operazione di caricamento. Come carattere sostitutivo puoi specificare qualsiasi carattere ASCII valido tranne NULL. Il carattere sostitutivo predefinito è un punto interrogativo (?). COPY sostituisce i caratteri multibyte con una stringa sostitutiva della stessa lunghezza. Ad esempio, un carattere di 4 byte viene sostituito da `'????'`.

COPIA restituisce il numero di righe che contenevano caratteri UTF-8 non validi. Aggiunge inoltre una voce alla tabella di sistema STL_REPLACEMENTS per ogni riga interessata, fino a un massimo di

100 righe per sezione di nodo. Vengono sostituiti anche altri caratteri UTF-8 non validi, ma gli eventi di sostituzione non vengono registrati.

ACCEPTINVCHARS è valido solo per le colonne VARCHAR.

Per questa fase, aggiungi ACCEPTINVCHARS con il carattere sostitutivo `'^'`.

MANIFEST

Quando si esegue una copia da Amazon S3 con COPY utilizzando un prefisso della chiave, esiste il rischio di caricare tabelle non desiderate. Ad esempio, la cartella `'s3://mybucket/load/` contiene otto file di dati che condividono il prefisso di chiave `customer-fw.tbl`: `customer-fw.tbl0000`, `customer-fw.tbl0001` e così di seguito. Tuttavia, la stessa cartella contiene anche file estranei, ovvero `customer-fw.tbl.log` e `customer-fw.tbl-0001.bak`.

Per essere certo di caricare tutti i file corretti e solo quelli, utilizza un file manifest. Il manifest è un file di testo in formato JSON che elenca esplicitamente la chiave oggetto univoca di ogni file di origine da caricare. Gli oggetti file possono essere in cartelle o in bucket differenti, ma nella stessa regione. Per ulteriori informazioni, consulta MANIFEST.

Quanto segue mostra il testo `customer-fw-manifest`.

```
{
  "entries": [
    {"url":"s3://<your-bucket-name>/load/customer-fw.tbl-000"},
    {"url":"s3://<your-bucket-name>/load/customer-fw.tbl-001"},
    {"url":"s3://<your-bucket-name>/load/customer-fw.tbl-002"},
    {"url":"s3://<your-bucket-name>/load/customer-fw.tbl-003"},
    {"url":"s3://<your-bucket-name>/load/customer-fw.tbl-004"},
    {"url":"s3://<your-bucket-name>/load/customer-fw.tbl-005"},
    {"url":"s3://<your-bucket-name>/load/customer-fw.tbl-006"},
    {"url":"s3://<your-bucket-name>/load/customer-fw.tbl-007"}
    ]
}
```

Per caricare i dati per la tabella CUSTOMER mediante un file manifest

1. Aprire il file `customer-fw-manifest` in un editor di testo.
2. Sostituire *<your-bucket-name>* con il nome del bucket.
3. Salvare il file.

4. Caricare il file nella cartella di caricamento del bucket.
5. Esegui il seguente comando COPY.

```
copy customer from 's3://<your-bucket-name>/load/customer-fw-manifest'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
fixedwidth 'c_custkey:10, c_name:25, c_address:25, c_city:10, c_nation:15,
 c_region :12, c_phone:15,c_mktsegment:10'
maxerror 10
acceptinvchars as '^'
manifest;
```

Caricamento della tabella DWDATE mediante DATEFORMAT

In questa fase, utilizzi le opzioni DELIMITER and DATEFORMAT per caricare la tabella DWDATE.

Durante il caricamento di colonne DATE e TIMESTAMP, COPY prevede che i dati siano nel formato predefinito, ovvero AAAA-MM-GG per le date e AAAA-MM-GG HH:MM:SS per i timestamp. Se i dati di caricamento non utilizzano il formato predefinito, puoi utilizzare DATEFORMAT e TIMEFORMAT per specificare il formato.

L'estratto seguente mostra i formati di data nella tabella DWDATE. Nota la discordanza tra i formati di data nella seconda colonna.

```
19920104 1992-01-04          Sunday  January 1992 199201 Jan1992 1 4 4 1...
19920112 January 12, 1992 Monday  January 1992 199201 Jan1992 2 12 12 1...
19920120 January 20, 1992 Tuesday     January 1992 199201 Jan1992 3 20 20 1...
```

DATEFORMAT

Puoi specificare un solo formato di data. Se i dati di caricamento contengono formati discordanti in varie colonne o se il formato non è noto al momento del caricamento, utilizza DATEFORMAT con l'argomento `'auto'`. Quando `'auto'` è specificato, COPY riconoscerà qualsiasi formato di data o ora valido e lo convertirà nel formato predefinito. L'opzione `'auto'` riconosce diversi formati che non sono supportati quando si utilizza una stringa DATEFORMAT e TIMEFORMAT. Per ulteriori informazioni, consulta Utilizzo del riconoscimento automatico con DATEFORMAT e TIMEFORMAT.

Per caricare la tabella DWDATE, esegui il comando COPY seguente.

```
copy dwdate from 's3://<your-bucket-name>/load/dwdate-tab.tbl'
```

```
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
delimiter '\t'
dateformat 'auto';
```

Caricamento della tabella LINEORDER mediante molteplici file

Questa fase utilizza le opzioni GZIP e COMPUPDATE per caricare la tabella LINEORDER.

In questo esercizio, carichi la tabella LINEORDER da un singolo file di dati e quindi la carichi di nuovo da molteplici file. In questo modo puoi confrontare i tempi di caricamento dei due metodi.

> **Note**
>
> I file per il caricamento della tabella LINEORDER sono forniti in un bucket AWS di esempio. Non devi caricare i file per questa fase.

GZIP, LZOP e BZIP2

Puoi comprimere i tuoi file utilizzando i formati di compressione gzip, lzop o bzip2. Nel caso di caricamento da file compressi, COPY decomprime i file durante il processo di caricamento. La compressione di file consente di risparmiare spazio di storage e riduce i tempi di caricamento.

COMPUPDATE

Quando COPY carica una tabella vuota senza codifiche di compressione, analizza i dati di caricamento per determinare le codifiche ottimali. Modifica quindi la tabella per utilizzare tali codifiche prima dell'inizio del caricamento. Questo processo di analisi richiede tempo, ma è necessario al massimo una sola volta per tabella. Per risparmiare tempo, puoi ignorare questa fese disattivando COMPUPDATE. Per consentire una valutazione accurata dei tempi di copia con il comando COPY, disattivi COMPUPDATE per questa fase.

Molteplici file

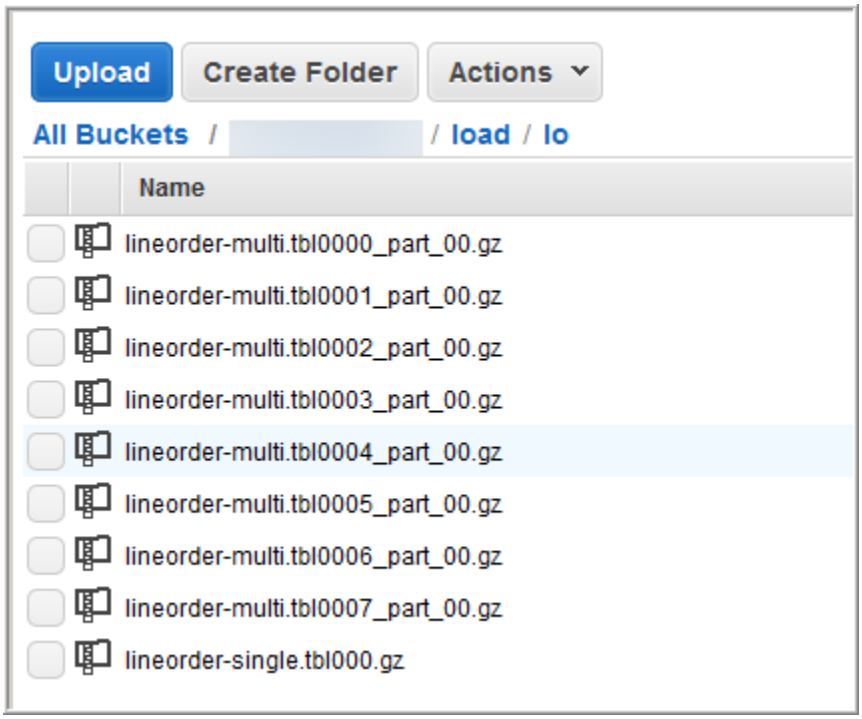
Il comando COPY consente caricamenti di dati estremamente efficaci quando vengono eseguiti da molteplici file in parallelo anziché da un singolo file. Dividi i dati in file in modo che il numero di file sia un multiplo del numero di sezioni nel cluster. Facendolo, Amazon Redshift divide il carico di lavoro e distribuisce i dati in modo omogeneo tra le sezioni del cluster. Il numero di sezioni per nodo dipende dalla dimensione dei nodi del cluster. Per ulteriori informazioni sul numero di sezioni per

ogni dimensione di nodo, consulta Informazioni su cluster e nodi nella Guida alla gestione di Amazon Redshift.

Ad esempio, i nodi di calcolo dc2.large utilizzati in questo tutorial hanno due sezioni ciascuno, di conseguenza un cluster a quattro nodi ha in totale otto sezioni. Nelle fasi precedenti, i dati di caricamento erano contenuti in otto file, anche se i file erano molto piccoli. In questa fase, confronti la differenza di tempo tra il caricamento da un singolo file voluminoso e quello da più file.

I file che utilizzi per questo tutorial contengono all'incirca 15 milioni di record e occupano intorno a 1,2 GB. Questi file sono molto piccoli nel contesto di Amazon Redshift, ma sono sufficienti per dimostrare i vantaggi prestazionali inerenti al caricamento da molteplici file. I file sono hanno dimensioni sufficientemente grandi perché il tempo necessario per scaricarli e quindi caricarli in Amazon S3 risulti eccessivo per questo tutorial. Pertanto, si caricano i file direttamente da un bucket di AWS esempio.

La screenshot seguente mostra i file di dati per LINEORDER.



Per valutare le prestazioni di COPY con molteplici file

1. Eseguire il comando COPY seguente per copiare da un singolo file. Non modificare il nome del bucket.

```
copy lineorder from 's3://awssampledb/load/lo/lineorder-single.tbl'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
```

```
gzip
compupdate off
region 'us-east-1';
```

2. I risultati dovrebbero essere simili a quanto mostrato di seguito. Notare il tempo di esecuzione.

```
Warnings:
Load into table 'lineorder' completed, 14996734 record(s) loaded successfully.

0 row(s) affected.
copy executed successfully

Execution time: 51.56s
```

3. Eseguire il comando COPY seguente per copiare da molteplici file. Non modificare il nome del bucket.

```
copy lineorder from 's3://awssampledb/load/lo/lineorder-multi.tbl'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
gzip
compupdate off
region 'us-east-1';
```

4. I risultati dovrebbero essere simili a quanto mostrato di seguito. Notare il tempo di esecuzione.

```
Warnings:
Load into table 'lineorder' completed, 14996734 record(s) loaded successfully.

0 row(s) affected.
copy executed successfully

Execution time: 17.7s
```

5. Confrontare i tempi di esecuzione.

   Nell'esempio, il tempo per caricare 15 milioni di record è sceso da 51,56 secondi a 17,7 secondi; una riduzione del 65,7%.

   Questi risultati sono basati sull'utilizzo di un cluster a quattro nodi. Se il cluster utilizzato ha più nodi, il risparmio di tempo sarà ancora maggiore. Per i cluster Amazon Redshift tipici, con decine di migliaia di nodi, la differenza sarà ancora più spettacolare. Se invece si dispone di un cluster a nodo singolo, la differenza tra i tempi di esecuzione sarà minima.

Approfondimenti

Fase 6: vacuum e analisi del database

Fase 6: vacuum e analisi del database

Quando aggiungi, elimini o modifichi un numero significativo di righe, devi eseguire un comando VACUUM e quindi il comando ANALYZE. Un comando VACUUM recupera lo spazio dalle righe eliminate e ripristina l'ordinamento. Il comando ANALYZE aggiorna i metadati delle statistiche, consentendo all'ottimizzatore di query di generare piani di query più accurati. Per ulteriori informazioni, consulta Vacuum delle tabelle.

Se carichi i dati in base all'ordine delle chiavi di ordinamento, un vacuum è alquanto veloce. In questo tutorial, hai aggiunto un numero importante di righe, ma le hai aggiunte a delle tabelle vuote. Inoltre, non hai eliminato delle righe, di conseguenza non è necessario eseguire un riordinamento. COPY aggiorna automaticamente le statistiche dopo aver caricato una tabella vuota, quindi le tue statistiche dovrebbero esserlo up-to-date. Tuttavia, per una corretta manutenzione, completi questo tutorial eseguendo un'operazione di vacuum e analizzando il database.

Per eseguire un vacuum e analizzare il database, esegui i comandi seguenti.

```
vacuum;
analyze;
```

Approfondimenti

Fase 7: elimina le risorse

Fase 7: elimina le risorse

Il tuo cluster genera dei costi fino a che è in esecuzione. Una volta completato questo tutorial, sarà quindi necessario ripristinare lo stato precedente dell'ambiente seguendo le istruzioni contenute nella Fase 5: Revoca dell'accesso ed eliminazione del cluster di esempio in Guida alle operazioni di base di Amazon Redshift.

Se vuoi conservare il cluster, ma recuperare lo storage utilizzato dalle tabelle SSB, esegui i comandi seguenti.

```
drop table part;
drop table supplier;
```

```
drop table customer;
drop table dwdate;
drop table lineorder;
```

Next

Riepilogo

Riepilogo

In questo tutorial, sono stati caricati file di dati in Amazon S3 e quindi sono stati utilizzati i comandi COPY per caricare dati di file nelle tabelle Amazon Redshift.

Hai caricato i dati utilizzando i seguenti formati:

- Delimitato da carattere
- CSV
- A larghezza fissa

Hai utilizzato la tabella di sistema STL_LOAD_ERRORS e quindi le opzioni REGION, MANIFEST, MAXERROR, ACCEPTINVCHARS, DATEFORMAT e NULL AS per correggere gli errori di caricamento.

Hai applicato le seguenti best practice per il caricamento dei dati:

- Utilizzo di un comando COPY per il caricamento dei dati
- Caricamento di file di dati
- Utilizzo di un singolo comando COPY per il caricamento da più file
- Compressione dei file di dati
- Verifica dei file di dati prima e dopo un caricamento

Per ulteriori informazioni sulle best practice di Amazon Redshift, fare riferimento ai seguenti collegamenti:

- Best practice di Amazon Redshift per il caricamento di dati
- Best practice di Amazon Redshift per la progettazione di tabelle
- Best practice di Amazon Redshift per la progettazione di query

Scaricamento dei dati

Argomenti

Per scaricare i dati dalle tabelle di database in un set di file in un bucket Amazon S3, è possibile usare il comando UNLOAD con un'istruzione SELECT. Puoi scaricare i dati di testo in formato delimitato o a larghezza fissa, indipendentemente dal formato usato per caricare i dati. Puoi anche specificare se creare file GZIP compressi.

È possibile limitare l'accesso degli utenti al bucket Amazon S3 usando credenziali di sicurezza temporanee.

Scaricamento dei dati in Amazon S3

Amazon Redshift suddivide i risultati di un'istruzione SELECT in un set di file, tra uno o più file per sezione del nodo, per semplificare il ricaricamento parallelo dei dati. In alternativa, puoi specificare che UNLOAD deve scrivere i risultati in modo seriale in uno o più file aggiungendo l'opzione PARALLEL OFF. Puoi limitare le dimensioni dei file in Amazon S3 specificando il parametro MAXFILESIZE. UNLOAD esegue automaticamente la crittografia dei file di dati usando la crittografia lato server di Amazon S3 (SSE-S3).

Nel comando UNLOAD è possibile usare qualsiasi istruzione SELECT supportata da Amazon Redshift, ad eccezione di un'istruzione SELECT che usa la clausola LIMIT nella parte esterna. Puoi ad esempio usare un'istruzione SELECT che include colonne specifiche o che usa una clausola WHERE per unire più tabelle. Se la query contiene virgolette (ad esempio per racchiudere valori letterali), è necessario farle precedere da un carattere di escape (\') nel testo della query. Per ulteriori informazioni, consultare le informazioni di riferimento per il comando SELECT. Per ulteriori informazioni sull'uso di una clausola LIMIT, consultare le Note per l'utilizzo per il comando UNLOAD.

Il comando UNLOAD seguente invia, ad esempio, i contenuti della tabella VENUE al bucket Amazon S3 `s3://mybucket/tickit/unload/`.

```
unload ('select * from venue')
to 's3://mybucket/tickit/unload/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

I nomi dei file creati nell'esempio precedente includono il prefisso "". 'venue_'.

```
venue_0000_part_00
venue_0001_part_00
venue_0002_part_00
venue_0003_part_00
```

Per impostazione predefinita, UNLOAD scrive i dati in parallelo in più file, in base al numero di sezioni nel cluster. Per scrivere i dati in un singolo file, specifica PARALLEL OFF. UNLOAD scrive i dati in modo seriale, con un ordinamento assoluto in base alla clausola ORDER BY, se viene usata. Le dimensioni massime per un file di dati sono di 6,2 GB. Se le dimensioni dei dati superano il limite massimo, UNLOAD crea file aggiuntivi con dimensioni fino a 6,2 GB ognuno.

L'esempio seguente scrive i contenuti di VENUE in un singolo file. È richiesto un solo file perché le sue dimensioni sono inferiori a 6,2 GB.

```
unload ('select * from venue')
to 's3://mybucket/tickit/unload/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
parallel off;
```

> **Note**
>
> Il comando UNLOAD è progettato per usare l'elaborazione parallela. Nella maggior parte dei casi, è consigliabile lasciare abilitata l'opzione PARALLEL, in particolare se i file verranno usati per caricare le tabelle usando un comando COPY.

Presupponendo che le dimensioni totali di VENUE siano di 5 GB, l'esempio seguente scrive i contenuti di VENUE in 50 file, ognuno di dimensioni di 100 MB.

```
unload ('select * from venue')
to 's3://mybucket/tickit/unload/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
parallel off
```

```
maxfilesize 100 mb;
```

Se si include un prefisso nella stringa del percorso Amazon S3, UNLOAD userà tale prefisso per i nomi di file.

```
unload ('select * from venue')
to 's3://mybucket/tickit/unload/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Puoi creare un file manifest che elenca i file scaricati specificando l'opzione MANIFEST nel comando UNLOAD. Il manifest è un file di testo in formato JSON che elenca in modo esplicito l'URL di ciascun file scritto in Amazon S3.

L'esempio seguente include l'opzione MANIFEST.

```
unload ('select * from venue')
to 's3://mybucket/tickit/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
manifest;
```

L'esempio seguente mostra un manifest per quattro file scaricati.

```
{
  "entries": [
    {"url":"s3://mybucket/tickit/venue_0000_part_00"},
    {"url":"s3://mybucket/tickit/venue_0001_part_00"},
    {"url":"s3://mybucket/tickit/venue_0002_part_00"},
    {"url":"s3://mybucket/tickit/venue_0003_part_00"}
  ]
}
```

Il file manifest può essere usato per caricare gli stessi file usando un comando COPY con l'opzione MANIFEST. Per ulteriori informazioni, consultare Utilizzo di un manifest per specificare i fili di dati.

Dopo aver completato un'operazione UNLOAD, verificare che i dati siano stati scaricati correttamente passando al bucket Amazon S3 in cui UNLOAD ha scritto i file. Vedrai uno o più file numerati per sezione, a partire dal numero zero. Se hai specificato l'opzione MANIFEST, sarà presente anche un file che termina con "". 'manifest'. Ad esempio:

```
mybucket/tickit/venue_0000_part_00
mybucket/tickit/venue_0001_part_00
```

```
mybucket/tickit/venue_0002_part_00
mybucket/tickit/venue_0003_part_00
mybucket/tickit/venue_manifest
```

È possibile ottenere un elenco dei file scritti su Amazon S3 a livello di programmazione chiamando un'operazione di elenco Amazon S3 dopo il completamento di UNLOAD. È inoltre possibile eseguire una query su STL_UNLOAD_LOG.

La query seguente restituisce il percorso per i file creati da un comando UNLOAD. La funzione PG_LAST_QUERY_ID restituisce la query più recente.

```
select query, substring(path,0,40) as path
from stl_unload_log
where query=2320
order by path;

query |             path
-------+---------------------------------------
  2320 | s3://my-bucket/venue0000_part_00
  2320 | s3://my-bucket/venue0001_part_00
  2320 | s3://my-bucket/venue0002_part_00
  2320 | s3://my-bucket/venue0003_part_00
(4 rows)
```

Se la quantità dei dati è molto elevata, Amazon Redshift può suddividere i file in più parti per sezione. Ad esempio:

```
venue_0000_part_00
venue_0000_part_01
venue_0000_part_02
venue_0001_part_00
venue_0001_part_01
venue_0001_part_02
...
```

Il comando UNLOAD seguente include una stringa tra virgolette nell'istruzione SELECT, pertanto per le virgolette viene eseguito l'escape (=\'0H\' ').

```
unload ('select venuename, venuecity from venue where venuestate=\'0H\' ')
to 's3://mybucket/tickit/venue/ '
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Per impostazione predefinita, UNLOAD ha esito negativo se deve sovrascrivere i file esistenti nel bucket di destinazione. Per sovrascrivere i file esistenti, incluso il file manifest, specifica l'opzione ALLOWOVERWRITE.

```
unload ('select * from venue')
to 's3://mybucket/venue_pipe_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
manifest
allowoverwrite;
```

Scaricamento di file di dati crittografati

Il comando UNLOAD crea automaticamente i file usando la crittografia lato server di Amazon S3 con chiavi di crittografia gestite da AWS (SSE-S3). Puoi anche specificare la crittografia lato server con una chiave AWS Key Management Service (SSE-KMS) oppure la crittografia lato client con una chiave gestita dal cliente (CSE-CMK). UNLOAD non supporta la crittografia lato server di Amazon S3 con una chiave gestita dal cliente. Per ulteriori informazioni, consultare Protezione dei dati con la crittografia lato server.

Per scaricare i dati in Amazon S3 usando la crittografia lato server con una chiave AWS KMS, utilizzare il parametro KMS_KEY_ID per fornire l'ID della chiave, come illustrato nell'esempio seguente.

```
unload ('select venuename, venuecity from venue')
to 's3://mybucket/encrypted/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
KMS_KEY_ID '1234abcd-12ab-34cd-56ef-1234567890ab'
encrypted;
```

Se si desidera fornire una chiave di crittografia personalizzata, è possibile creare file di dati con crittografia lato client in Amazon S3 usando il comando UNLOAD con l'opzione ENCRYPTED. UNLOAD usa lo stesso processo di crittografia envelope usato dalla crittografia lato client di Amazon S3. Puoi quindi usare il comando COPY con l'opzione ENCRYPTED per caricare i file crittografati.

Il processo avviene in questo modo:

1. Devi creare una chiave AES a 256 bit con codifica base64 da usare come chiave di crittografia privata o chiave simmetrica root.
2. Invia un comando UNLOAD che include la chiave simmetrica root e l'opzione ENCRYPTED.

3. UNLOAD genera una chiave simmetrica una tantum (denominata chiave simmetrica envelope) e un vettore di inizializzazione (IV), che usa per crittografare i dati.
4. UNLOAD esegue la crittografia della chiave simmetrica envelope usando la chiave simmetrica root.
5. UNLOAD archivia quindi i file di dati crittografati in Amazon S3 e archivia la chiave envelope crittografata e il vettore di inizializzazione come metadati degli oggetti con ogni file. La chiave envelope crittografata viene archiviata come metadati degli oggetti `x-amz-meta-x-amz-key` e il vettore di inizializzazione viene archiviato come metadati degli oggetti `x-amz-meta-x-amz-iv`.

Per ulteriori informazioni sul processo di crittografia envelope, consultare l'articolo Crittografia dei dati lato client con l'SDK AWS per Java e Amazon S3.

Per scaricare i file di dati crittografati, aggiungi il valore della chiave root alla stringa di credenziali e includi l'opzione ENCRYPTED. Se usi l'opzione MANIFEST, viene crittografato anche il file manifest.

```
unload ('select venuename, venuecity from venue')
to 's3://mybucket/encrypted/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
master_symmetric_key '<root_key>'
manifest
encrypted;
```

Per scaricare file di dati crittografati compressi in formato GZIP, includi l'opzione GZIP insieme al valore della chiave root e all'opzione ENCRYPTED.

```
unload ('select venuename, venuecity from venue')
to 's3://mybucket/encrypted/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
master_symmetric_key '<root_key>'
encrypted gzip;
```

Per caricare i file di dati crittografati, aggiungi il parametro MASTER_SYMMETRIC_KEY con lo stesso valore della chiave root e includi l'opzione ENCRYPTED.

```
copy venue from 's3://mybucket/encrypted/venue_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
master_symmetric_key '<root_key>'
encrypted;
```

Scaricamento dei dati in formato delimitato o a larghezza fissa

Puoi scaricare i dati in formato delimitato o a larghezza fissa. L'output predefinito è delimitato da barra verticale (usando il carattere "|").

L'esempio seguente specifica la virgola come delimitatore:

```
unload ('select * from venue')
to 's3://mybucket/tickit/venue/comma'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter ',';
```

I file di output risultanti avranno l'aspetto seguente:

```
20,Air Canada Centre,Toronto,ON,0
60,Rexall Place,Edmonton,AB,0
100,U.S. Cellular Field,Chicago,IL,40615
200,Al Hirschfeld Theatre,New York City,NY,0
240,San Jose Repertory Theatre,San Jose,CA,0
300,Kennedy Center Opera House,Washington,DC,0
...
```

Per scaricare lo stesso set di risultati in un file delimitato da tabulazioni, usa il comando seguente:

```
unload ('select * from venue')
to 's3://mybucket/tickit/venue/tab'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter as '\t';
```

In alternativa, puoi usare una specifica FIXEDWIDTH. Questa specifica è costituita da un identificatore per ogni colonna di tabella e dalla larghezza della colonna (numero di caratteri). Il comando UNLOAD ha esito negativo se deve troncare i dati, quindi specifica una larghezza almeno equivalente alla voce più lunga per la colonna. Lo scaricamento di dati a larghezza fissa è analogo a quello di dati delimitati, ad eccezione del fatto che l'output risultante non contiene caratteri di delimitazione. Ad esempio:

```
unload ('select * from venue')
to 's3://mybucket/tickit/venue/fw'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
fixedwidth '0:3,1:100,2:30,3:2,4:6';
```

L'output a larghezza fissa ha l'aspetto seguente:

```
20 Air Canada Centre         Toronto      ON0
60 Rexall Place              Edmonton     AB0
100U.S. Cellular Field       Chicago      IL40615
200Al Hirschfeld Theatre     New York CityNY0
240San Jose Repertory TheatreSan Jose     CA0
300Kennedy Center Opera HouseWashington   DC0
```

Per ulteriori informazioni sulle specifiche FIXEDWIDTH, consultare il comando UNLOAD.

Nuovo caricamento dei dati scaricati

Per ricaricare i risultati di un'operazione di scaricamento, puoi usare un comando COPY.

L'esempio seguente mostra un caso semplice in cui la tabella VENUE viene scaricata usando un file manifest, troncata e ricaricata.

```
unload  ('select * from venue order by venueid')
to 's3://mybucket/tickit/venue/reload_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
manifest
delimiter '|';

truncate venue;

copy venue
from 's3://mybucket/tickit/venue/reload_manifest'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
manifest
delimiter '|';
```

Dopo il ricaricamento, l'aspetto della tabella VENUE è simile al seguente:

```
select * from venue order by venueid limit 5;

 venueid |         venuename          |  venuecity  | venuestate | venueseats
---------+----------------------------+-------------+------------+-----------
       1 | Toyota Park                | Bridgeview  | IL         |          0
       2 | Columbus Crew Stadium      | Columbus    | OH         |          0
```

```
       3 | RFK Stadium               | Washington  | DC         |          0
       4 | CommunityAmerica Ballpark | Kansas City | KS         |          0
       5 | Gillette Stadium          | Foxborough  | MA         |      68756
(5 rows)
```

Creazione di funzioni definite dall'utente

Puoi creare una funzione scalare definita dall'utente personalizzata usando una clausola SQL SELECT o un programma Python. La nuova funzione è archiviata nel database ed è disponibile per qualsiasi utente con privilegi sufficienti per l'esecuzione. Una funzione definita dall'utente scalare personalizzata viene definita nello stesso modo in cui vengono eseguite le funzioni Amazon Redshift esistenti.

Per le funzioni definite dall'utente Python, oltre ad aggiungere la funzionalità Python standard, puoi importare moduli Python personalizzati. Per ulteriori informazioni, consulta Supporto del linguaggio Python per funzioni definite dall'utente. Nota che Python 3 non è disponibile per le UDF di Python. Per ottenere il supporto di Python 3 per le UDF di Amazon Redshift, usa invece. Creazione di una funzione Lambda definita dall'utente scalare

Puoi anche creare AWS Lambda UDF che utilizzano funzioni personalizzate definite in Lambda come parte delle tue query SQL. Le funzioni Lambda definite dall'utente consentono di scrivere funzioni definite dall'utente complesse e di integrarle con componenti di terze parti. Possono anche consentire di superare alcune delle limitazioni delle funzioni definite dall'utente Python e SQL correnti. Ad esempio, possono aiutare ad accedere alle risorse di rete e di archiviazione e a scrivere istruzioni SQL più complete. È possibile creare UDF Lambda in uno qualsiasi dei linguaggi di programmazione supportati da Lambda, come Java, Go, Node.js, C# PowerShell, Python e Ruby. Oppure è possibile usare un runtime personalizzato.

Per impostazione predefinita, tutti gli utenti possono eseguire funzioni definite dall'utente. Per ulteriori informazioni sui privilegi, consultare Sicurezza e privilegi dell'UDF.

Argomenti

Sicurezza e privilegi dell'UDF

Per creare una UDF devi disporre dell'autorizzazione per l'utilizzo nel linguaggio per SQL o plpythonu (Python). Per impostazione predefinita, USAGE ON LANGUAGE SQL viene concesso a PUBLIC, ma devi concedere esplicitamente USAGE ON LANGUAGE PLPYTHONU a utenti o gruppi specifici.

Per revocare l'utilizzo per SQL, revoca innanzitutto l'utilizzo da PUBLIC. Quindi concedi l'utilizzo su SQL solo a utenti o gruppi specifici autorizzati a creare UDF SQL. L'esempio seguente revoca l'utilizzo su SQL da PUBLIC. Concede quindi l'utilizzo al gruppo di utenti `udf_devs`.

```
revoke usage on language sql from PUBLIC;
grant usage on language sql to group udf_devs;
```

Per eseguire una funzione definita dall'utente, è necessario disporre dell'autorizzazione di esecuzione per ciascuna funzione. Per impostazione predefinita, l'autorizzazione per eseguire nuove funzioni definite dall'utente è concessa a PUBLIC. Per limitare l'utilizzo, revocare l'autorizzazione da PUBLIC. Quindi concedi il privilegio a specifici individui o gruppi.

L'esempio seguente revoca l'esecuzione su una funzione `f_py_greater` da PUBLIC. Concede quindi l'utilizzo al gruppo di utenti `udf_devs`.

```
revoke execute on function f_py_greater(a float, b float) from PUBLIC;
grant execute on function f_py_greater(a float, b float) to group udf_devs;
```

Per impostazione predefinita gli utenti con privilegi avanzati hanno tutti i privilegi.

Per ulteriori informazioni, consultare GRANT e REVOKE.

Creazione di una funzione definita dall'utente SQL scalare

Una funzione definita dall'utente SQL scalare integra una clausola SQL SELECT che viene eseguita quando la funzione viene chiamata e restituisce un singolo valore. Il comando CREATE FUNCTION definisce i parametri seguenti:

- Argomenti di input (facoltativi). Ogni argomento deve avere un tipo di dati.
- Un tipo di dati restituito.
- Una clausola SQL SELECT. Nella clausola SELECT fai riferimento agli argomenti di input usando $1, $2 e così via, in base all'ordine degli argomenti nella definizione della funzione.

Il tipo di dati di input e il tipo di dati restituito possono essere qualsiasi tipo di dati di Amazon Redshift standard.

Non includere una clausola FROM nella clausola SELECT. Includi invece la clausola FROM nell'istruzione SQL che chiama la funzione definita dall'utente SQL.

La clausola SELECT non può includere nessuno dei seguenti tipi di clausole:

- FROM
- INTO
- WHERE
- GROUP BY
- ORDER BY
- LIMIT

Esempio di funzione SQL scalare

L'esempio seguente crea una funzione che confronta due numeri e restituisce il valore più grande. Per ulteriori informazioni, consultare CREATE FUNCTION.

```
create function f_sql_greater (float, float)
  returns float
stable
as $$
  select case when $1 > $2 then $1
    else $2
  end
$$ language sql;
```

La query seguente chiama la nuova funzione f_sql_greater per eseguire una query sulla tabella SALES e restituire il valore di COMMISSION o il 20% di PRICEPAID, a seconda di quale valore è più grande.

```
select f_sql_greater(commission, pricepaid*0.20) from sales;
```

Denominazione delle funzioni definite dall'utente

Puoi evitare possibili conflitti e risultati imprevisti valutando con attenzione le convenzioni di denominazione delle funzioni definite dall'utente prima dell'implementazione. Poiché i nomi delle funzioni possono essere soggetti a overload, possono entrare in conflitto con nomi di funzioni Amazon Redshift esistenti e future. Questo argomento descrive l'overload e presenta una strategia per evitare i conflitti.

Overload dei nomi delle funzioni

Una funzione è identificata dal proprio nome e da una firma, che corrisponde al numero di argomenti di input e ai tipi di dati degli argomenti. Due funzioni nello stesso schema possono avere lo stesso nome se hanno firme diverse. In altri termini, i nomi delle funzioni possono essere in overload.

Quando si esegue una query, il motore di query determina quale funzione chiamare in base al numero di argomenti specificati e ai tipi di dati degli argomenti. È possibile usare l'overload per simulare funzioni con un numero variabile di argomenti, fino al limite consentito dal comando CREATE FUNCTION.

Prevenzione dei conflitti di denominazione delle funzioni definite dall'utente

Si consiglia di assegnare un nome a tutte le funzioni definite dall'utente utilizzando il prefisso `f_`. Amazon Redshift riserva il prefisso `f_` esclusivamente per le funzioni definite dall'utente e aggiungendo il prefisso `f_` ai nomi delle funzioni dell'utente, si evita che il nome della funzione entri in conflitto con altri nomi di funzioni SQL predefinite di Amazon Redshift correnti o future. Ad esempio, assegnando a una nuova funzione definita dall'utente `f_sum`, si evita il conflitto con la funzione SUM di Amazon Redshift. Analogamente, se si assegna il nome `f_fibonacci` a una nuova funzione, è possibile evitare un conflitto se Amazon Redshift aggiungerà una funzione denominata FIBONACCI in una versione futura.

È possibile creare una funzione definita dall'utente con lo stesso nome e la stessa firma di una funzione SQL predefinita di Amazon Redshift esistente senza che avvenga l'overload del nome della funzione se la funzione definita dall'utente e la funzione predefinita si trovano in schemi diversi. Poiché le funzioni predefinite si trovano nello schema del catalogo di sistema pg_catalog, puoi creare una funzione definita dall'utente con lo stesso nome in un altro schema, ad esempio uno schema pubblico o definito dall'utente. In alcuni casi, è possibile chiamare una funzione non qualificata esplicitamente con un nome di schema. In tal caso, Amazon Redshift cerca prima lo schema pg_catalog per impostazione predefinita. Pertanto, una funzione incorporata viene eseguita prima di una nuova funzione definita dall'utente con lo stesso nome.

È possibile modificare questo comportamento impostando il percorso di ricerca in modo da aggiungere pg_catalog alla fine. In questo caso, le funzioni definite dall'utente avranno la precedenza su quelle predefinite, ma è possibile che si verifichino dei risultati imprevisti. L'adozione di una strategia di denominazione univoca, ad esempio usando il prefisso riservato `f_`, è un approccio più affidabile. Per ulteriori informazioni, consultare SET e search_path.

Creazione di una funzione definita dall'utente Python scalare

Una funzione definita dall'utente Python integra un programma Python che viene eseguito quando la funzione viene chiamata e restituisce un singolo valore. Il comando CREATE FUNCTION definisce i parametri seguenti:

- Argomenti di input (facoltativi). Ogni argomento deve avere un nome e un tipo di dati.
- Un tipo di dati restituito.
- Un programma Python eseguibile.

Il tipo di dati di input e restituiti può essere SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE o TIMESTAMP. Inoltre, le funzioni definite dall'utente Python possono usare il tipo di dati ANYELEMENT, convertito automaticamente da Amazon Redshift in un tipo di dati standard in base agli argomenti specificati in fase di runtime. Per ulteriori informazioni, consultare Tipo di dati ANYELEMENT

Quando una query di Amazon Redshift chiama una funzione definita dall'utente scalare, si verifica quanto segue in fase di runtime.

1. La funzione converte gli argomenti di input in tipi di dati Python.

   Per una mappatura dei tipi di dati di Amazon Redshift a tipi di dati Python, consultare Tipi di dati delle funzioni definite dall'utente Python.
2. La funzione esegue il programma Python, passando gli argomenti di input convertiti.
3. Il codice Python restituisce un singolo valore. Il tipo di dati del valore restituito deve corrispondere al tipo di dati RETURNS specificato dalla definizione della funzione.
4. La funzione converte il valore restituito da Python nel tipo di dati di Amazon Redshift specificato e quindi restituisce il valore alla query.

> **Note**
>
> Python 3 non è disponibile per le UDF Python. Per ottenere il supporto di Python 3 per le UDF di Amazon Redshift, usa invece. Creazione di una funzione Lambda definita dall'utente scalare

Esempio di funzione definita dall'utente Python scalare

L'esempio seguente crea una funzione che confronta due numeri e restituisce il valore più grande. Tenere presente che il rientro del codice tra i segni di doppio dollaro ($$) è un requisito di Python. Per ulteriori informazioni, consultare CREATE FUNCTION.

```
create function f_py_greater (a float, b float)
  returns float
stable
as $$
  if a > b:
    return a
  return b
$$ language plpythonu;
```

La query seguente chiama la nuova funzione `f_greater` per eseguire una query sulla tabella SALES e restituire il valore di COMMISSION o il 20% di PRICEPAID, a seconda di quale valore è più grande.

```
select f_py_greater (commission, pricepaid*0.20) from sales;
```

Tipi di dati delle funzioni definite dall'utente Python

Le funzioni definite dall'utente Python possono usare qualsiasi tipo di dati di Amazon Redshift standard per gli argomenti di input e il valore restituito della funzione. Oltre ai tipi di dati standard, le funzioni definite dall'utente supportano il tipo di dati ANYELEMENT, convertito automaticamente da Amazon Redshift in un tipo di dati standard in base agli argomenti specificati in fase di runtime. Le funzioni definite dall'utente scalari possono restituire un tipo di dati ANYELEMENT. Per ulteriori informazioni, consultare Tipo di dati ANYELEMENT.

Durante l'esecuzione, Amazon Redshift converte gli argomenti da tipi di dati Amazon Redshift a tipi di dati Python per l'elaborazione. Quindi converte il valore restituito dal tipo di dati Python al tipo di

dati Amazon Redshift corrispondente. Per ulteriori informazioni sui tipi di dati di Amazon Redshift, consultare Tipi di dati.

La tabella seguente indica la mappatura dei tipi di dati Amazon Redshift ai tipi di dati Python.

Tipo di dati di Amazon Redshift	Tipo di dati Python
smallint	int
integer	
bigint	
short	
Long	
decimal o numeric	decimal
double	float
real	
booleano	bool
char	Stringa
varchar	
timestamp	datetime

Tipo di dati ANYELEMENT

ANYELEMENT è un tipo di dati polimorfico. Ciò significa che se una funzione viene dichiarata usando ANYELEMENT come tipo di dati di un argomento, la funzione può accettare qualsiasi tipo di dati Amazon Redshift standard come input per l'argomento quando viene chiamata. L'argomento ANYELEMENT viene impostato sul tipo di dati effettivamente passato all'argomento quando viene chiamata la funzione.

Se una funzione usa più tipi di dati ANYELEMENT, questi devono tutti essere risolti nello stesso tipo di dati effettivo quando viene chiamata la funzione. Tutti i tipi di dati dell'argomento ANYELEMENT

sono impostati sull'effettivo tipo di dati del primo argomento passato a ANYELEMENT. Ad esempio, una funzione dichiarata come `f_equal(anyelement, anyelement)` accetta uno qualsiasi tra due valori di input, purché abbiano lo stesso tipo di dati.

Se il valore restituito di una funzione viene dichiarato come ANYELEMENT, almeno un argomento di input deve essere ANYELEMENT. Il tipo di dati effettivo per il valore restituito sarà lo stesso del tipo di dati effettivo specificato per l'argomento di input ANYELEMENT.

Supporto del linguaggio Python per funzioni definite dall'utente

Puoi creare una funzione definita dall'utente personalizzata basata sul linguaggio di programmazione Python. La libreria standard Python 2.7 può essere usata in funzioni definite dall'utente, con l'eccezione dei moduli seguenti:

- ScrolledText
- Tix
- Tkinter
- tk
- turtle
- smtpd

Oltre alla libreria standard Python, i moduli seguenti fanno parte dell'implementazione di Amazon Redshift:

- numpy 1.8.2
- pandas 0.14.1
- python-dateutil 2.2
- pytz 2014.7
- scipy 0.12.1
- six 1.3.0
- wsgiref 0.1.2

Puoi anche importare moduli Python personalizzati e renderli disponibili per l'uso in funzioni definite dall'utente eseguendo un comando CREATE LIBRARY. Per ulteriori informazioni, consultare Importazione di moduli di libreria Python personalizzati.

Important

Amazon Redshift blocca tutto l'accesso di rete e l'accesso in scrittura al file system tramite funzioni definite dall'utente.

Note

Python 3 non è disponibile per le UDF Python. Per ottenere il supporto di Python 3 per le UDF di Amazon Redshift, usa invece. Creazione di una funzione Lambda definita dall'utente scalare

Importazione di moduli di libreria Python personalizzati

Puoi definire funzioni scalari usando la sintassi del linguaggio Python. È possibile utilizzare i moduli di libreria Python standard e i moduli Amazon Redshift preinstallati. Inoltre è possibile creare moduli di libreria Python personalizzati e importare le librerie nei cluster oppure usare librerie esistenti da Python o da terze parti.

Non è possibile creare una libreria che contiene un modulo con lo stesso nome di un modulo della libreria standard Python o un modulo Python preinstallato in Amazon Redshift. Se una libreria installata dall'utente esistente usa lo stesso pacchetto Python di una libreria creata da te, devi eliminare la libreria esistente prima di installare quella nuova.

Per installare librerie personalizzate, devi essere un utente con privilegi avanzati o avere il privilegio `USAGE ON LANGUAGE plpythonu`. Tuttavia, qualsiasi utente con privilegi sufficienti per la creazione di funzioni può usare le librerie installate. Puoi eseguire query sul catalogo di sistema PG_LIBRARY per visualizzare informazioni sulle librerie installate nel cluster.

Per importare un modulo Python personalizzato nel cluster

Questa sezione presenta un esempio di importazione di un modulo Python personalizzato nel cluster. Per completare la procedura in questa sezione, è necessario disporre di un bucket Amazon S3 in cui caricare il pacchetto della libreria. Devi quindi installare il pacchetto nel cluster. Per ulteriori informazioni sulla creazione di un bucket, consultare Creazione di un bucket nella Guida per l'utente di Amazon Simple Storage Service.

In questo esempio, supponi di aver creato funzioni definite dall'utente per usarle con posizioni e distanze nei dati. Connettersi al cluster Amazon Redshift da uno strumento client SQL ed eseguire i comandi seguenti per creare le funzioni.

```
CREATE FUNCTION f_distance (x1 float, y1 float, x2 float, y2 float) RETURNS float
 IMMUTABLE as $$
    def distance(x1, y1, x2, y2):
        import math
        return math.sqrt((y2 - y1) ** 2 + (x2 - x1) ** 2)

    return distance(x1, y1, x2, y2)
$$ LANGUAGE plpythonu;

CREATE FUNCTION f_within_range (x1 float, y1 float, x2 float, y2 float) RETURNS bool
 IMMUTABLE as $$
    def distance(x1, y1, x2, y2):
        import math
        return math.sqrt((y2 - y1) ** 2 + (x2 - x1) ** 2)

    return distance(x1, y1, x2, y2) < 20
$$ LANGUAGE plpythonu;
```

Come puoi notare, nell'esempio precedente alcune righe di codice sono duplicate. Queste righe duplicate sono necessarie, perché una funzione definita dell'utente non può fare riferimento al contenuto di un'altra funzione definita dall'utente ed entrambe le funzioni devono avere le stesse funzionalità. Tuttavia, invece di duplicare il codice in più funzioni, puoi creare una libreria personalizzata e configurare le funzioni perché la usino.

A questo scopo, crea prima di tutto il pacchetto della libreria completando questa procedura:

1. Creare una cartella denominata geometry. Questa cartella è il pacchetto di primo livello della libreria.
2. Nella cartella geometry creare un file denominato `__init__.py`. Notare che il nome del file contiene due caratteri di sottolineatura doppi. Questo file indica a Python che il pacchetto può essere inizializzato.
3. Nella cartella geometry creare una cartella denominata trig. Questa cartella è il sottopacchetto della libreria.
4. Nella cartella trig creare un altro file denominato `__init__.py` e un file denominato `line.py`. In questa cartella `__init__.py` indica a Python che il sottopacchetto può essere inizializzato e che `line.py` è il file che contiene il codice della libreria.

La struttura dei file e delle cartelle deve essere uguale alla seguente:

```
geometry/
   __init__.py
   trig/
      __init__.py
      line.py
```

Per ulteriori informazioni sulla struttura dei pacchetti, consultare la pagina relativa ai moduli nel tutorial su Python nel sito Web Python.

5. Il codice seguente contiene una classe e funzioni membro per la libreria. Copiare e incollare il codice in `line.py`.

```
class LineSegment:
  def __init__(self, x1, y1, x2, y2):
    self.x1 = x1
    self.y1 = y1
    self.x2 = x2
    self.y2 = y2
  def angle(self):
    import math
    return math.atan2(self.y2 - self.y1, self.x2 - self.x1)
  def distance(self):
    import math
    return math.sqrt((self.y2 - self.y1) ** 2 + (self.x2 - self.x1) ** 2)
```

Dopo aver creato il pacchetto, completare le operazioni seguenti per prepararlo e caricarlo in Amazon S3.

1. Comprimere il contenuto della cartella geometry in un file ZIP denominato geometry.zip. Non includere la cartella geometry stessa, ma solo il contenuto della cartella, come mostrato di seguito:

```
geometry.zip
   __init__.py
   trig/
      __init__.py
      line.py
```

2. Caricare geometry.zip sul bucket Amazon S3.

> **Important**
>
> Se il bucket Amazon S3 non si trova nella stessa regione del cluster Amazon Redshift, è necessario utilizzare l'opzione REGION per specificare la regione in cui si trovano i dati. Per ulteriori informazioni, consultare CREATE LIBRARY.

3. Dallo strumento client SQL eseguire il comando seguente per installare la libreria. <bucket_name>Sostituiscilo con il nome del tuo bucket e sostituiscilo <access key id><secret key>con una chiave di accesso e una chiave di accesso segreta ricavate dalle tue credenziali utente AWS Identity and Access Management (IAM).

```
CREATE LIBRARY geometry LANGUAGE plpythonu FROM 's3://<bucket_name>/geometry.zip'
 CREDENTIALS 'aws_access_key_id=<access key id>;aws_secret_access_key=<secret key>';
```

Dopo aver installato la libreria nel cluster, devi configurare le funzioni per l'uso della libreria. A questo scopo, esegui i comandi seguenti.

```
CREATE OR REPLACE FUNCTION f_distance (x1 float, y1 float, x2 float, y2 float) RETURNS
 float IMMUTABLE as $$
    from trig.line import LineSegment

    return LineSegment(x1, y1, x2, y2).distance()
$$ LANGUAGE plpythonu;

CREATE OR REPLACE FUNCTION f_within_range (x1 float, y1 float, x2 float, y2 float)
 RETURNS bool IMMUTABLE as $$
    from trig.line import LineSegment

    return LineSegment(x1, y1, x2, y2).distance() < 20
$$ LANGUAGE plpythonu;
```

Nei comandi precedenti `import trig/line` elimina il codice duplicato dalle funzioni originali in questa sezione. Puoi riutilizzare la funzionalità fornita da questa libreria in più funzioni definite dall'utente. Tieni presente che per importare il modulo, devi solo specificare il percorso del sottopacchetto e il nome del modulo (`trig/line`).

Vincoli delle funzioni definite dall'utente

Entro i vincoli indicati in questo argomento, è possibile usare funzioni definite dall'utente per tutti i casi in cui si usano le funzioni scalari predefinite di Amazon Redshift. Per ulteriori informazioni, consultare Informazioni di riferimento sulle funzioni SQL.

Alle funzioni definite dall'utente Python di Amazon Redshift si applicano i vincoli seguenti:

- Le funzioni definite dall'utente Python non possono accedere alla rete né leggere o scrivere nel file system.
- Le dimensioni totali delle librerie Python installate dall'utente non possono superare 100 MB.
- Il numero di funzioni definite dall'utente Python che possono essere eseguite contemporaneamente per ogni cluster sono limitate a un quarto del livello di simultaneità totale per il cluster. Ad esempio, se il cluster è configurato con simultaneità 15, può essere eseguito un massimo di tre funzioni definite dall'utente contemporaneamente. Una volta raggiunto il limite, le funzioni definite dall'utente vengono accodate per l'esecuzione all'interno di code di gestione dei carichi di lavoro. Alle funzioni definite dall'utente SQL non si applica alcun limite di simultaneità. Per ulteriori informazioni, consulta Implementazione della gestione del carico di lavoro.
- Quando si utilizzano UDF Python, Amazon Redshift non supporta i tipi di dati SUPER e HLLSKETCH.

Logging di errori e avvisi in funzioni definite dall'utente

Puoi usare il modulo di logging Python per creare messaggi di errore e avviso definiti dall'utente nelle funzioni definite dall'utente. In seguito all'esecuzione della query, puoi eseguire una query sulla vista di sistema SVL_UDF_LOG per recuperare i messaggi registrati.

> **Note**
>
> Il logging di funzioni definite dall'utente utilizza risorse del cluster e può influire sulle prestazioni del sistema. Ti consigliamo di implementare il logging solo per scopi di sviluppo e risoluzione dei problemi.

Durante l'esecuzione delle query, il gestore dei log scrive messaggi nella vista di sistema SVL_UDF_LOG, insieme a nome, nodo e sezione della funzione corrispondente. Il gestore dei log scrive una riga nella vista di sistema SVL_UDF_LOG per ogni messaggio e per ogni sezione. I

messaggi vengono troncati a 4096 byte. I log delle funzioni definite dall'utente è limitato a 500 righe per sezione. Quando il log è pieno, il gestore dei log elimina i messaggi meno recenti e aggiunge un messaggio di avviso in SVL_UDF_LOG.

> **Note**
>
> Il gestore dei log delle funzioni definite dall'utente di Amazon Redshift aggiunge a caratteri di nuova riga (\n), caratteri di barra verticale (|) e caratteri di barra rovesciata (\) un carattere di barra rovesciata (\) per l'escape.

Per impostazione predefinita, il livello di log delle funzioni definite dall'utente è impostato su WARNING. I messaggi con livello di log WARNING, ERROR e CRITICAL vengono registrati. I messaggi con gravità minore, come INFO, DEBUG e NOTSET, vengono ignorati. Per impostare il livello di log delle funzioni definite dall'utente, usa il metodo logger Python. Ad esempio, il codice seguente imposta il livello di log su INFO.

```
logger.setLevel(logging.INFO)
```

Per ulteriori informazioni sull'uso del modulo di logging Python, consultare la pagina relativa all'utilità di logging per Python nella documentazione di Python.

L'esempio seguente crea una funzione denominata f_pyerror che importa il modulo di logging Python, crea un'istanza del logger e registra un errore.

```
CREATE OR REPLACE FUNCTION f_pyerror()
RETURNS INTEGER
VOLATILE AS
$$
import logging

logger = logging.getLogger()
logger.setLevel(logging.INFO)
logger.info('Your info message here')
return 0
$$ language plpythonu;
```

L'esempio seguente esegue una query su SVL_UDF_LOG per visualizzare il messaggio registrato nell'esempio precedente.

```
select funcname, node, slice, trim(message) as message
from svl_udf_log;

  funcname  | query | node | slice |   message
------------+-------+------+-------+-----------------
  f_pyerror | 12345 |     1|     1 | Your info message here
```

Creazione di una funzione Lambda definita dall'utente scalare

Amazon Redshift può utilizzare funzioni personalizzate definite AWS Lambda come parte delle query SQL. È possibile scrivere UDF Lambda scalari in qualsiasi linguaggio di programmazione supportato da Lambda, ad esempio Java, Go, Node.js, C# PowerShell, Python e Ruby. Oppure è possibile usare un runtime personalizzato.

Le funzioni Lambda definite dall'utente sono definite e gestite in Lambda ed è possibile controllare i privilegi di accesso per richiamare queste funzioni definite dall'utente in Amazon Redshift. È possibile richiamare più funzioni Lambda nella stessa query o richiamare la stessa funzione più volte.

Utilizzare le funzioni Lambda definite dall'utente in tutte le clausole delle istruzioni SQL in cui sono supportate le funzioni scalari. Le funzioni Lambda definite dall'utente possono essere utilizzate in qualsiasi istruzione SQL, ad esempio SELECT, UPDATE, INSERT o DELETE.

> **Note**
>
> L'utilizzo di funzioni Lambda definite dall'utente può comportare costi aggiuntivi dal servizio Lambda. Ciò dipende da fattori quali il numero di richieste Lambda (richiami alle funzioni definite dall'utente) e la durata totale dell'esecuzione del programma Lambda. Tuttavia, non sono previsti costi aggiuntivi per l'utilizzo delle funzioni Lambda definite dall'utente in Amazon Redshift. Per informazioni sui prezzi di AWS Lambda, consulta AWS Lambda Prezzi.
> Il numero di richieste Lambda varia a seconda della clausola di istruzione SQL specifica in cui viene utilizzata la funzioni Lambda definite dall'utente. Si supponga, ad esempio, che la funzione sia utilizzata in una clausola WHERE come la seguente.
> `SELECT a, b FROM t1 WHERE lambda_multiply(a, b) = 64; SELECT a, b FROM t1 WHERE a*b = lambda_multiply(2, 32)`
> In questo caso, Amazon Redshift chiama la prima istruzione SELECT per ogni clausola e chiama la seconda istruzione SELECT una sola volta.

Tuttavia, l'utilizzo di una funzione definita dall'utente nella parte di proiezione della query potrebbe richiamare la funzione Lambda una sola volta per ogni riga qualificata o aggregata nel set di risultati.

Registrazione di una funzione Lambda definita dall'utente

Il comando CREATE EXTERNAL FUNCTION crea i seguenti parametri:

- (Facoltativo) Un elenco di argomenti con tipo di dati.
- Un tipo di dati restituito.
- Un nome di funzione della funzione esterna chiamata da Amazon Redshift.
- Un ruolo IAM che il cluster Amazon Redshift è autorizzato ad assumere e chiamare Lambda.
- Il nome di una funzione Lambda richiamato dalla funzione Lambda definita dall'utente.

Per informazioni su CREATE EXTERNAL FUNCTION, consultare CREATE EXTERNAL FUNCTION.

I tipi di dati di input e di restituzione per questa funzione possono essere qualsiasi tipo di dati Amazon Redshift standard.

Amazon Redshift assicura che la funzione esterna possa inviare e ricevere argomenti e risultati in batch.

Gestione della sicurezza e dei privilegi della funzione Lambda definita dall'utente

Per creare una funzione Lambda definita dall'utente, assicurati di disporre delle autorizzazioni per l'utilizzo su LANGUAGE EXFUNC. È necessario concedere o revocare esplicitamente USE ON LANGUAGE EXFUNC a utenti, gruppi o pubblici specifici.

Nell'esempio seguente viene concesso l'utilizzo su EXFUNC a PUBLIC.

```
grant usage on language exfunc to PUBLIC;
```

Nell'esempio seguente viene revocato l'utilizzo su exfunc da PUBLIC, quindi viene concesso al gruppo di utenti lambda_udf_devs.

```
revoke usage on language exfunc from PUBLIC;
```

```
grant usage on language exfunc to group lambda_udf_devs;
```

Per eseguire una funzione Lambda definita dall'utente, assicurarsi di disporre delle autorizzazioni per ogni funzione chiamata. Per impostazione predefinita, l'autorizzazione per eseguire nuove funzioni Lambda definite dall'utente è concessa a PUBLIC. Per limitare l'utilizzo, revocare l'autorizzazione da PUBLIC. Quindi concedere il privilegio a individui o gruppi specifici.

Nell'esempio seguente viene revocata l'esecuzione su una funzione exfunc_sum da PUBLIC. Concede quindi l'utilizzo al gruppo di utenti lambda_udf_devs.

```
revoke execute on function exfunc_sum(int, int) from PUBLIC;
grant execute on function exfunc_sum(int, int) to group lambda_udf_devs;
```

Per impostazione predefinita gli utenti con privilegi avanzati hanno tutti i privilegi.

Per ulteriori informazioni su come concedere e revocare privilegi, consultare GRANT e REVOKE.

Configurazione del parametro di autorizzazione per le funzioni Lambda definite dall'utente

Il comando CREATE EXTERNAL FUNCTION richiede l'autorizzazione per richiamare le funzioni Lambda AWS Lambda. Per avviare l'autorizzazione, specifica un ruolo AWS Identity and Access Management (IAM) quando esegui il comando CREATE EXTERNAL FUNCTION. Per ulteriori informazioni sui ruoli IAM, consultare Ruoli IAM nella Guida per l'utente di IAM.

Se è presente un ruolo IAM esistente per richiamare funzioni Lambda collegate al cluster, è possibile sostituire l'Amazon Resource Name (ARN) del ruolo nel parametro IAM_ROLE per il comando. Nelle sezioni seguenti vengono descritti i passaggi per l'uso di un ruolo IAM nel comando CREATE EXTERNAL FUNCTION.

Creazione di un ruolo IAM per Lambda

Il ruolo IAM richiede l'autorizzazione per richiamare le funzioni Lambda. Durante la creazione del ruolo IAM, fornire l'autorizzazione in uno dei modi seguenti:

- Collegare la policy `AWSLambdaRole` sulla pagina Collega policy di autorizzazioni durante la creazione di un ruolo IAM. La policy `AWSLambdaRole` concede le autorizzazioni per richiamare funzioni Lambda, che è il requisito minimo. Per ulteriori informazioni e altre policy, consultare Policy IAM basate sull'identità per AWS Lambda nella Guida per gli sviluppatori di AWS Lambda .

- Creare policy personalizzate da associare al ruolo IAM con l'autorizzazione `lambda:InvokeFunction` di tutte le risorse o di una particolare funzione Lambda con l'ARN di tale funzione. Per informazioni sulla creazione di policy IAM, consultare Creazione di policy IAM nella Guida per l'utente di IAM.

  La seguente policy di esempio consente di richiamare Lambda su una particolare funzione Lambda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Invoke",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": "arn:aws:lambda:us-west-2:123456789012:function:my-function"
        }
    ]
}
```

  Per ulteriori informazioni sulle risorse per le funzioni Lambda, consultare Risorse e condizioni per operazioni Lambda in Documentazione di riferimento dell'API IAM.

  Dopo aver creato la policy personalizzata con le autorizzazioni richieste, è possibile collegare la policy al ruolo IAM sulla pagina Collega policy di autorizzazioni durante la creazione di un ruolo IAM.

Per i passaggi per creare un ruolo IAM, consulta Autorizzazione di Amazon Redshift ad accedere ad AWS altri servizi per tuo conto nella Amazon Redshift Management Guide.

Se non si desidera creare un nuovo ruolo IAM, è possibile aggiungere le autorizzazioni menzionate in precedenza al ruolo IAM esistente.

Associazione di un ruolo IAM al cluster

Collegare il ruolo IAM al cluster. È possibile aggiungere un ruolo a un cluster o visualizzare i ruoli associati a un cluster utilizzando la Console di gestione, la CLI o l'API di Amazon Redshift. Per

ulteriori informazioni, consulta Associazione di un ruolo IAM a un cluster nella Guida alla gestione di Amazon Redshift.

Inclusione del ruolo IAM nel comando

Includere l'ARN del ruolo IAM nel comando CREATE EXTERNAL FUNCTION. Quando crei un ruolo IAM, IAM restituisce un Amazon Resource Name (ARN) per il ruolo. Per specificare un ruolo IAM, fornire l'ARN del ruolo con il parametro `IAM_ROLE`. Di seguito è mostrata la sintassi del parametro `IAM_ROLE`.

```
IAM_ROLE 'arn:aws:iam::aws-account-id:role/role-name'
```

Per richiamare le funzioni Lambda che risiedono in altri account all'interno della stessa regione, consultare Concatenazione di ruoli IAM in Amazon Redshift.

Utilizzo dell'interfaccia JSON tra Amazon Redshift e AWS Lambda

Amazon Redshift utilizza un'interfaccia comune per tutte le funzioni Lambda con cui Amazon Redshift comunica.

La tabella seguente mostra l'elenco dei campi di input che le funzioni Lambda designate possono aspettarsi per il payload JSON.

Nome campo	Descrizione	Intervallo di valori
request_id	Un identificatore universale univoco (UID, Universally Unique Identifier) che identifichi in modo univoco ogni richiesta di richiamo.	Un UUID valido.
cluster	L'Amazon Resource Name (ARN) del cluster database.	Un ARN del cluster valido.

Nome campo	Descrizione	Intervallo di valori
Utente	Il nome dell'utente che effettua la chiamata.	Un nome utente valido.
database	Il nome del database su cui è in esecuzione la query.	Un nome valido del database.
funzione external_	Il nome completo della funzione esterna che effettua la chiamata.	Un nome di funzione completo valido.
query_id	L'ID query della query che effettua la chiamata.	Un ID query valido.
num_records	Il numero di argomenti nel payload.	Un valore compreso tra 1 e 2^64.
argomenti	Il payload di dati nel formato specificato.	I dati in formato array devono essere un array JSON. Ogni elemento è un record che è un array se il numero di argomenti è maggiore di 1. Utilizzando un array, Amazon Redshift mantiene l'ordine dei record nel payload.

L'ordine dell'array JSON determina l'ordine di elaborazione batch. La funzione Lambda deve elaborare gli argomenti in maniera iterativa e produrre il numero esatto di record. Di seguito è riportato un esempio di payload.

```
{
  "request_id" : "23FF1F97-F28A-44AA-AB67-266ED976BF40",
  "cluster" : "arn:aws:redshift:xxxx",
  "user" : "adminuser",
  "database" : "db1",
  "external_function": "public.foo",
  "query_id" : 5678234,
  "num_records" : 4,
```

```
  "arguments" : [
     [ 1, 2 ],
     [ 3, null],
     null,
     [ 4, 6]
   ]
 }
```

L'output restituito dalla funzione Lambda contiene i seguenti campi.

Nome campo	Descrizione	Intervallo di valori
success	L'indicazione di successo o fallimento per la funzione.	Un valore di `"true"` o `"false"`.
error_msg	Il messaggio di errore se il valore di riuscita è `"false"` (se la funzione non riesce); in caso contrario, questo campo viene ignorato.	Un messaggio valido.
num_records	Il numero di record nel payload.	Un valore compreso tra 1 e 2^64.
results	I risultati della chiamata nel formato specificato.	N/D

Di seguito è riportato un esempio di output di una funzione Lambda:

```
{
  "success": true,   // true indicates the call succeeded
  "error_msg" : "my function isn't working",  // shall only exist when success != true
  "num_records": 4,      // number of records in this payload
  "results" : [
     1,
     4,
```

```
      null,
      7
    ]
}
```

Quando vengono richiamate le funzioni Lambda da query SQL, Amazon Redshift garantisce la sicurezza della connessione con le seguenti considerazioni:

- Autorizzazioni GRANT e REVOKE. Per ulteriori informazioni sulla sicurezza e sui privilegi delle funzioni definite dall'utente, consultare Sicurezza e privilegi dell'UDF.
- Amazon Redshift invia alla funzione Lambda designata solo il set minimo di dati.
- Amazon Redshift chiama solo la funzione Lambda designata con il ruolo IAM designato.

Esempi di utilizzo di funzioni definite dall'utente (UDF, user-defined function)

Le funzioni definite dall'utente possono essere utilizzate per risolvere i problemi aziendali integrando Amazon Redshift con altri componenti. Di seguito sono riportati alcuni esempi di come altri utenti hanno utilizzato le UDF per i loro casi d'uso:

- Accesso a componenti esterni utilizzando le UDF Lambda di Amazon Redshift: descrive come funzionano le UDF Lambda di Amazon Redshift e come creare una UDF Lambda.
- Traduzione e analisi del testo tramite le funzioni SQL con Amazon Redshift, Amazon Translate e Amazon Comprehend: fornisce UDF Lambda di Amazon Redshift predefinite che è possibile installare con pochi clic per tradurre, modificare e analizzare i campi di testo.
- Accesso ad Amazon Location Service da Amazon Redshift: descrive come utilizzare le UDF Lambda di Amazon Redshift per l'integrazione con Amazon Location Service.
- Tokenizzazione dei dati con Amazon Redshift e Protegrity: descrive come integrare le UDF Lambda di Amazon Redshift con il prodotto Protegrity Serverless.
- UDF di Amazon Redshift: una raccolta delle UDF di Amazon Redshift SQL, Lambda e Python.

Creazione di procedure archiviate in Amazon Redshift

Una procedura archiviata di Amazon Redshift può essere definita utilizzando il linguaggio procedurale PostgreSQL PL/pgSQL per eseguire un set di query SQL e operazioni logiche. La procedura è archiviata nel database ed è disponibile per qualsiasi utente con privilegi sufficienti.

A differenza di una funzione definita dall'utente (UDF), una procedura archiviata può incorporare il linguaggio di definizione dei dati (DDL) e il linguaggio di manipolazione dei dati (DML) oltre alle query SELECT. Una procedura archiviata non deve restituire un valore. Puoi includere il linguaggio procedurale, compreso il looping e le espressioni condizionali, per controllare il flusso logico.

Per informazioni sui comandi SQL per creare e gestire le procedura archiviate, consultare i seguenti argomenti sui comandi:

- CREATE PROCEDURE
- ALTER PROCEDURE
- DROP PROCEDURE
- SHOW PROCEDURE
- CALL
- GRANT
- REVOKE
- ALTER DEFAULT PRIVILEGES

Argomenti

Panoramica delle procedure archiviate in Amazon Redshift

Le procedure archiviate vengono comunemente utilizzate per integrare la logica di trasformazione dei dati, quella specifica aziendale nonché la convalida dei dati. Combinando più passaggi di SQL in una procedura archiviata, è possibile ridurre i cicli tra le applicazioni e il database.

Per il controllo granulare degli accessi, è possibile creare procedure archiviate per eseguire funzioni senza consentire all'utente di accedere alle tabelle sottostanti. Ad esempio, solo il proprietario

o l'utente con privilegi avanzati può troncare una tabella e a un utente serve l'autorizzazione di scrittura per inserire dati in una tabella. Invece di concedere a un utente le autorizzazioni nella tabelle sottostanti, puoi creare una procedura archiviata che esegua l'operazione. In seguito fornisci l'autorizzazione all'utente per eseguire la procedura archiviata.

Una procedura archiviata con l'attributo di sicurezza DEFINER viene eseguita con i privilegi del proprietario della procedura archiviata. Per impostazione predefinita, una procedura archiviata ha sicurezza INVOKER, che significa che la procedura utilizza i privilegi dell'utente che richiama la procedura.

Per creare una procedura archiviata, utilizza il comando CREATE PROCEDURE. Per eseguire una procedura, utilizza il comando CALL. Più avanti in questa sezione sono presenti degli esempi.

Note

Alcuni client possono visualizzare il seguente errore durante la creazione di una procedura archiviata di Amazon Redshift.

```
ERROR: 42601: [Amazon](500310) unterminated dollar-quoted string at or near "$$
```

Questo errore si verifica a causa dell'incapacità del client eseguire correttamente il parsing dell'istruzione CREATE PROCEDURE con punto e virgola che delimita le istruzioni e con il segno del dollaro ($). In questo caso solo una parte dell'istruzione viene inviata al server Amazon Redshift. Spesso si può risolvere questo problema utilizzando l'opzione `Run as batch` o `Execute selected` del client.
Ad esempio, quando si utilizza un client Aginity, utilizzare l'opzione `Run entire script as batch`. Quando si utilizza SQL Workbench/J, consigliamo la versione 124. Quando si utilizza SQL Workbench/J versione 125, è possibile specificare un delimitatore alternativo come soluzione alternativa.
CREATE PROCEDURE contiene istruzioni SQL delimitate da un punto e virgola (;). Definire un delimitatore alternativo come una barra (/) e posizionarlo alla fine dell'istruzione CREATE PROCEDURE per inviare l'istruzione al server Amazon Redshift per l'elaborazione. Di seguito è riportato un esempio.

```
CREATE OR REPLACE PROCEDURE test()
AS $$
BEGIN
  SELECT 1 a;
END;
```

```
$$
LANGUAGE plpgsql
;
/
```

Per ulteriori informazioni, consultare Delimitatore alternativo nella documentazione SQL Workbench/J. Oppure usa un client con un supporto migliore per l'analisi delle istruzioni CREATE PROCEDURE, come l'editor di query nella console Amazon Redshift o. TablePlus

Argomenti

- Denominazione delle stored procedure
- Sicurezza e privilegi per le procedure archiviate
- Restituzione di un set di risultati
- Gestione delle transazioni
- Errori di blocco
- Registrazione delle stored procedure
- Considerazioni per il supporto delle procedure archiviate

L'esempio seguente mostra una procedura senza argomenti di output. Come impostazione predefinita, gli argomenti sono argomenti di input (IN).

```
CREATE OR REPLACE PROCEDURE test_sp1(f1 int, f2 varchar)
AS $$
BEGIN
  RAISE INFO 'f1 = %, f2 = %', f1, f2;
END;
$$ LANGUAGE plpgsql;

call test_sp1(5, 'abc');
INFO: f1 = 5, f2 = abc
CALL
```

Note

Quando si scrivono le stored procedure, si consiglia di attenersi a una best practice per proteggere i valori sensibili:

Non eseguire la codifica fissa delle informazioni sensibili nella logica della procedura archiviata. Ad esempio, non assegnare una password utente in un'istruzione CREATE USER nel corpo di una procedura archiviata. Ciò rappresenta un rischio per la sicurezza, poiché i valori con codifica fissa possono essere registrati come metadati dello schema nelle tabelle del catalogo. È invece consigliabile passare i valori sensibili, ad esempio le password, come argomenti alla procedura archiviata, mediante parametri.
Per ulteriori informazioni sulle procedure archiviate, consulta CREATE PROCEDURE e Creazione di procedure archiviate in Amazon Redshift. Per ulteriori informazioni sulle tabelle di catalogo, consulta Tabelle di catalogo di sistema.

L'esempio seguente mostra una procedura con argomenti di output. Gli argomenti sono input (IN), input e output (INOUT), e output (OUT).

```
CREATE OR REPLACE PROCEDURE test_sp2(f1 IN int, f2 INOUT varchar(256), out_var OUT
 varchar(256))
AS $$
DECLARE
  loop_var int;
BEGIN
  IF f1 is null OR f2 is null THEN
    RAISE EXCEPTION 'input cannot be null';
  END IF;
  DROP TABLE if exists my_etl;
  CREATE TEMP TABLE my_etl(a int, b varchar);
    FOR loop_var IN 1..f1 LOOP
        insert into my_etl values (loop_var, f2);
        f2 := f2 || '+' || f2;
    END LOOP;
  SELECT INTO out_var count(*) from my_etl;
END;
$$ LANGUAGE plpgsql;


call test_sp2(2,'2019');

         f2          | column2
---------------------+---------
 2019+2019+2019+2019 | 2
(1 row)
```

Denominazione delle stored procedure

Se si definisce una procedura con lo stesso nome e tipi di dati di argomento di input o firma diversi, si crea una nuova procedura. Di conseguenza, il nome della procedura è sovraccarico. Per ulteriori informazioni, consulta Overload dei nomi delle procedure. Amazon Redshift non abilita l'overload della procedura in base ad argomenti di output. Non è possibile avere due procedure con lo stesso nome e tipi di dati di argomento di input ma tipi di argomento di output diversi.

Il proprietario o un utente con privilegi avanzati può sostituire il body di una procedura archiviata con una nuova con la stessa firma. Per modificare la forma o i tipi restituiti di una procedura archiviata, ignora la procedura archiviata e ricreala. Per ulteriori informazioni, consultare DROP PROCEDURE e CREATE PROCEDURE.

Puoi evitare possibili conflitti e risultati imprevisti valutando le convenzioni di denominazione delle procedure archiviate prima dell'implementazione. Poiché è possibile eseguire l'overload dei nomi delle procedure, è possibile che questi entrino in collisione con i nomi delle procedure Amazon Redshift esistenti e future.

Overload dei nomi delle procedure

Una procedura è identificata dal proprio nome e da una firma, che corrisponde al numero di argomenti di input e ai tipi di dati degli argomenti. Due procedure nello stesso schema possono avere lo stesso nome se hanno firme diverse. In altre parole, è possibile eseguire l'overload dei nomi delle procedure.

Quando si esegue una procedura, il motore di query determina quale procedura chiamare in base al numero di argomenti specificati e ai tipi di dati degli argomenti. Puoi usare l'overload per simulare procedure con un numero variabile di argomenti, fino al limite consentito dal comando CREATE PROCEDURE. Per ulteriori informazioni, consultare CREATE PROCEDURE.

Prevenzione dei conflitti di denominazione

Si consiglia di assegnare un nome a tutte le procedure utilizzando il prefisso `sp_`. Amazon Redshift riserva il prefisso `sp_` solo per le procedure archiviate. Aggiungendo il prefisso `sp_` ai nomi delle procedure, si assicura che il nome della procedura non sia in conflitto con alcuni nomi delle procedure Amazon Redshift esistenti o future.

Sicurezza e privilegi per le procedure archiviate

Per impostazione predefinita, tutti gli utenti hanno l'autorizzazione a creare una procedura. Per creare una procedura, è necessario il privilegio USAGE nel linguaggio PL/pgSQL, che viene concesso a PUBLIC per impostazione predefinita. Solo gli utenti con privilegi avanzati e i proprietari hanno per impostazione predefinita l'autorizzazione di richiamare una procedura. Gli utenti con privilegi avanzati possono eseguire REVOKE USAGE su PL/pgSQL da un utente se desiderano evitare che l'utente crei una procedura archiviata.

Per richiamare una procedura, è necessaria l'autorizzazione EXECUTE per la procedura. Per impostazione predefinita, il privilegio EXECUTE per le nuove procedure viene concessa al proprietario e agli utenti con privilegi avanzati della procedura. Per ulteriori informazioni, consulta GRANT.

L'utente che crea una procedura è il proprietario come impostazione predefinita. Il proprietario dispone di privilegi CREATE, DROP e EXECUTE sulla procedura come impostazione predefinita. Gli utenti con privilegi avanzati hanno tutti i privilegi.

L'attributo SECURITY controlla i privilegi di una procedura per accedere agli oggetti del database. Quando crei una procedura archiviata, puoi impostare l'attributo SECURITY su DEFINER o INVOKER. Se specifichi SECURITY INVOKER, la procedura utilizza i privilegi dell'utente che la chiama. Se specifichi SECURITY INVOKER, la procedura utilizza i privilegi del proprietario che la chiama. INVOKER è l'impostazione predefinita.

In quanto una procedura SECURITY DEFINER viene eseguita con i privilegi dell'utente proprietario della procedura, assicurarsi che la procedura non venga utilizzata impropriamente. Per assicurare che le procedure SECURITY DEFINER non vengano utilizzate impropriamente, esegui le seguenti operazioni:

- Concedi EXECUTE alle procedure SECURITY DEFINER per specificare gli utenti e non a PUBLIC.
- Qualificare tutti gli oggetti di database ai quali deve accedere la procedura con i nomi dello schema. Ad esempio, utilizza `myschema.mytable` invece di solo `mytable`.
- Se non puoi qualificare un nome di oggetto in base al suo schema, imposta `search_path` durante la creazione della procedura utilizzando l'opzione SET. Imposta `search_path` per escludere qualsiasi schema che è scrivibile da parte di utenti non attendibili. Questo approccio previene che qualsiasi intermediario di questa procedura crei oggetti (ad esempio, tabelle o visualizzazioni) che mascherano gli oggetti il cui scopo è di essere utilizzati dalla procedura. Per ulteriori informazioni sull'opzione SET, consultare CREATE PROCEDURE.

L'esempio seguente imposta `search_path` su `admin` per assicurare che l'accesso alla tabella `user_creds` avvenga dallo schema `admin` e non dal pubblico o da qualsiasi altro schema nel `search_path` dell'intermediario.

```
CREATE OR REPLACE PROCEDURE sp_get_credentials(userid int, o_creds OUT varchar)
AS $$
BEGIN
  SELECT creds INTO o_creds
  FROM user_creds
  WHERE user_id = $1;
END;
$$ LANGUAGE plpgsql
SECURITY DEFINER
-- Set a secure search_path
SET search_path = admin;
```

Restituzione di un set di risultati

Puoi restituire un set di risultati utilizzando un cursore o una tabella temporanea.

Restituzione di un cursore

Per restituire un cursore, crea una procedura con un argomento INOUT definito con un tipo di dati `refcursor`. Quando richiami la procedura, assegna un nome al cursore. È quindi possibile recuperare i risultati dal cursore in base al nome.

L'esempio seguente crea una procedura denominata `get_result_set` con un argomento INOUT denominato `rs_out` utilizzando il tipo di dati `refcursor`. La procedura apre il cursore utilizzando un'istruzione SELECT.

```
CREATE OR REPLACE PROCEDURE get_result_set (param IN integer, rs_out INOUT refcursor)
AS $$
BEGIN
  OPEN rs_out FOR SELECT * FROM fact_tbl where id >= param;
END;
$$ LANGUAGE plpgsql;
```

Il seguente comando CALL apre il cursore con il nome `mycursor`. Utilizza cursori solo nelle transazioni.

```
BEGIN;
```

```
CALL get_result_set(1, 'mycursor');
```

Dopo l'apertura del cursore, puoi recuperare dal cursore, come mostra l'esempio seguente.

```
FETCH ALL FROM mycursor;

    id | secondary_id | name
-------+--------------+---------
     1 |            1 | Joe
     1 |            2 | Ed
     2 |            1 | Mary
     1 |            3 | Mike
(4 rows)
```

Alla fine, la transazione viene confermata o ripristinata allo stato precedente.

```
COMMIT;
```

Un cursore restituito da una procedura archiviata è soggetto ai stessi vincoli e considerazioni di prestazione come descritto in DECLARE CURSOR. Per ulteriori informazioni, consultare Vincoli del cursore.

L'esempio seguente mostra la chiamata della procedura archiviata `get_result_set` mediante un tipo di dati `refcursor` da JDBC. Il `'mycursor'` letterale (il nome del cursore) viene passato a `prepareStatement`. In seguito, i risultati vengono recuperati da `ResultSet`.

```
static void refcursor_example(Connection conn) throws SQLException {
    conn.setAutoCommit(false);
    PreparedStatement proc = conn.prepareStatement("CALL get_result_set(1,
 'mycursor')");
    proc.execute();
    ResultSet rs = statement.executeQuery("fetch all from mycursor");
    while (rs.next()) {
      int n = rs.getInt(1);
      System.out.println("n " + n);
    }
```

Utilizzo di una tabella temporanea

Per restituire risultati, puoi restituire un handle a una tabella temporanea contenente righe di risultati. Il cliente può fornire un nome come un parametro alla procedura archiviata. All'interno della

procedura archiviata, l'SQL dinamico può essere utilizzato per funzionare nella tabella temporanea. Di seguito viene riportato un esempio.

```
CREATE PROCEDURE get_result_set(param IN integer, tmp_name INOUT varchar(256)) as $$
DECLARE
  row record;
BEGIN
  EXECUTE 'drop table if exists ' || tmp_name;
  EXECUTE 'create temp table ' || tmp_name || ' as select * from fact_tbl where id <= '
 || param;
END;
$$ LANGUAGE plpgsql;

CALL get_result_set(2, 'myresult');
 tmp_name
-----------
 myresult
(1 row)

SELECT * from myresult;
 id | secondary_id | name
----+--------------+------
  1 |            1 | Joe
  2 |            1 | Mary
  1 |            2 | Ed
  1 |            3 | Mike
(4 rows)
```

Gestione delle transazioni

Puoi creare una procedura archiviata con un comportamento predefinito di gestione delle transazioni o un comportamento non atomico.

Gestione delle transazioni con procedura archiviata in modalità predefinita

Il comportamento di commit automatico in modalità di transazione predefinita porta ogni comando SQL eseguito separatamente a eseguire il commit individualmente. Una chiamata a una procedura archiviata viene trattata come un comando SQL singolo. Le istruzioni SQL all'interno di una procedura si comportano come se fossero in un blocco di transazioni che inizia implicitamente quando la chiamata inizia e termina quando la chiamata finisce. Una chiamata nidificata a un'altra procedura viene trattata come qualsiasi altra istruzione SQL e opera nel contesto della stessa

transazione come intermediario. Per ulteriori informazioni sulla funzionalità WLM automatica, consultare Isolamento serializzabile.

Tuttavia, supponiamo di chiamare una procedura archiviata da un blocco di transazione specificata dall'utente (definito da BEGIN...COMMIT). In questo caso tutte le istruzioni nella procedura archiviata vengono eseguite nel contesto della transazione specificata dall'utente. La procedura non conferma implicitamente all'uscita. L'intermediario controlla la conferma della procedura o il ripristino allo stato precedente.

Se si verifica qualsiasi errore durante l'esecuzione di una procedura archiviata, tutte le modifiche apportate nella transazione corrente vengono ripristinate allo stato precedente.

Puoi utilizzare le seguenti istruzioni di controllo delle transazioni in una procedura archiviata:

- COMMIT: esegue il commit di tutto il lavoro fatto nella transazione attuale e avvia implicitamente una nuova transazione. Per ulteriori informazioni, consultare COMMIT.
- ROLLBACK: esegue il ripristino dello stato precedente del lavoro fatto nella transazione corrente e avvia implicitamente una nuova transazione. Per ulteriori informazioni, consultare ROLLBACK.

TRUNCATE è un'altra istruzione che è possibile avviare da una procedura archiviata e influenza la gestione delel transazioni. In Amazon Redshift, TRUNCATE emette un commit implicitamente. Il comportamento rimane lo stesso nel contesto delle procedure archiviate. Quando un'istruzione TRUNCATE viene emessa da una procedura archiviata, conferma la transazione corrente e inizia una nuova. Per ulteriori informazioni, consultare TRUNCATE.

Tutte le istruzioni che seguono un'istruzione COMMIT, ROLLBACK o TRUNCATE eseguita nel contesto di una nuova transazione. Agiscono in questo modo fino a quando incontrano un'istruzione COMMIT, ROLLBACK o TRUNCATE o si esce dalla procedura archiviata.

Quando si utilizza un'istruzione COMMIT, ROLLBACK o TRUNCATE da una procedura archiviata, vengono applicati i seguenti vincoli:

- Se la procedura archiviata viene chiamata da un blocco di transazione, non può emettere un'istruzione TRUNCATE, ROLLBACK o COMMIT. Questa restrizione si applica all'interno del body della procedura archiviata e all'interno di una chiamata di procedura nidificata.
- Se la procedura archiviata viene creata con le opzioni `SET config`, non può emettere un'istruzione COMMIT, ROLLBACK o TRUNCATE. Questa restrizione si applica all'interno del body della procedura archiviata e all'interno di una chiamata di procedura nidificata.

- Qualsiasi cursore che è aperto (esplicitamente o implicitamente) viene chiuso automaticamente quando viene elaborata un'istruzione TRUNCATE, COMMIT o ROLLBACK. Per i vincoli sui cursori espliciti e impliciti, consultaConsiderazioni per il supporto delle procedure archiviate.

Inoltre, non è possibile eseguire COMMIT o ROLLBACK utilizzando l'SQL dinamico. Tuttavia, puoi eseguire TRUNCATE tramite SQL dinamico. Per ulteriori informazioni, consultare SQL dinamico.

Quando si lavora con le procedure archiviate, considerare che le istruzioni BEGIN ed END in PL/pgSQL sono solo per i raggruppamenti. Non iniziano o terminano la transazione. Per ulteriori informazioni, consultare Blocco.

L'esempio seguente dimostra il comportamento delle transazioni quando si chiama una procedura archiviata dall'interno di un blocco di transazioni esplicite. Le due istruzioni insert emesse da al di fuori della procedura archiviata e quella dall'interno fanno tutte parte della stessa transazione (3382). La transazione viene confermata quando l'utente emette il commit esplicito.

```
CREATE OR REPLACE PROCEDURE sp_insert_table_a(a int) LANGUAGE plpgsql
AS $$
BEGIN
  INSERT INTO test_table_a values (a);
END;
$$;

Begin;
  insert into test_table_a values (1);
  Call sp_insert_table_a(2);
  insert into test_table_a values (3);
Commit;

select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;

 userid | xid  | pid |  type   |              stmt_text
--------+------+-----+---------+----------------------------------------
    103 | 3382 | 599 | UTILITY | Begin;
    103 | 3382 | 599 | QUERY   | insert into test_table_a values (1);
    103 | 3382 | 599 | UTILITY | Call sp_insert_table_a(2);
    103 | 3382 | 599 | QUERY   | INSERT INTO test_table_a values ( $1 )
    103 | 3382 | 599 | QUERY   | insert into test_table_a values (3);
    103 | 3382 | 599 | UTILITY | COMMIT
```

Al contrario, consideriamo un esempio quando le stesse istruzioni vengono emesse dal di fuori di un blocco di transazione esplicito e la sessione ha l'autocommit impostato su ON. In questo caso, ogni istruzione viene eseguita nella propria transazione.

```
insert into test_table_a values (1);
Call sp_insert_table_a(2);
insert into test_table_a values (3);

select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;

 userid | xid  | pid |  type   |
             stmt_text
--------+------+-----+---------
+-----------------------------------------------------------------------------------------------------
    103 | 3388 | 599 | QUERY   | insert into test_table_a values (1);
    103 | 3388 | 599 | UTILITY | COMMIT
    103 | 3389 | 599 | UTILITY | Call sp_insert_table_a(2);
    103 | 3389 | 599 | QUERY   | INSERT INTO test_table_a values ( $1 )
    103 | 3389 | 599 | UTILITY | COMMIT
    103 | 3390 | 599 | QUERY   | insert into test_table_a values (3);
    103 | 3390 | 599 | UTILITY | COMMIT
```

L'esempio seguente emette un'istruzione TRUNCATE dopo aver l'inserimento in `test_table_a`. L'istruzione TRUNCATE emette un commit implicito che conferma la transazione corrente (3335) e avvia una nuova (3336). La nuova transazione viene confermata all'uscita della procedura.

```
CREATE OR REPLACE PROCEDURE sp_truncate_proc(a int, b int) LANGUAGE plpgsql
AS $$
BEGIN
  INSERT INTO test_table_a values (a);
  TRUNCATE test_table_b;
  INSERT INTO test_table_b values (b);
END;
$$;

Call sp_truncate_proc(1,2);

select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;
```

```
 userid | xid  |  pid  |  type   |
                                        stmt_text
--------+------+-------+---------
+-----------------------------------------------------------------------------------------------
    103 | 3335 | 23636 | UTILITY | Call sp_truncate_proc(1,2);
    103 | 3335 | 23636 | QUERY   | INSERT INTO test_table_a values ( $1 )
    103 | 3335 | 23636 | UTILITY | TRUNCATE test_table_b
    103 | 3335 | 23636 | UTILITY | COMMIT
    103 | 3336 | 23636 | QUERY   | INSERT INTO test_table_b values ( $1 )
    103 | 3336 | 23636 | UTILITY | COMMIT
```

Il seguente esempio emette un'istruzione TRUNCATE da una chiamata nidificata. TRUNCATE conferma tutto il lavoro effettuato finora nelle procedura esterne e interne in una transazione (3344). Avvia una nuova transazione (3345). La nuova transazione viene confermata all'uscita della procedura esterna.

```
CREATE OR REPLACE PROCEDURE sp_inner(c int, d int) LANGUAGE plpgsql
AS $$
BEGIN
  INSERT INTO inner_table values (c);
  TRUNCATE outer_table;
  INSERT INTO inner_table values (d);
END;
$$;

CREATE OR REPLACE PROCEDURE sp_outer(a int, b int, c int, d int) LANGUAGE plpgsql
AS $$
BEGIN
  INSERT INTO outer_table values (a);
  Call sp_inner(c, d);
  INSERT INTO outer_table values (b);
END;
$$;

Call sp_outer(1, 2, 3, 4);

select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;

 userid | xid  |  pid  |  type   |
                                        stmt_text
```

```
--------+------+-------+---------
+-----------------------------------------------------------------------------------------------
    103 | 3344 | 23636 | UTILITY | Call sp_outer(1, 2, 3, 4);
    103 | 3344 | 23636 | QUERY   | INSERT INTO outer_table values ( $1 )
    103 | 3344 | 23636 | UTILITY | CALL sp_inner( $1 , $2 )
    103 | 3344 | 23636 | QUERY   | INSERT INTO inner_table values ( $1 )
    103 | 3344 | 23636 | UTILITY | TRUNCATE outer_table
    103 | 3344 | 23636 | UTILITY | COMMIT
    103 | 3345 | 23636 | QUERY   | INSERT INTO inner_table values ( $1 )
    103 | 3345 | 23636 | QUERY   | INSERT INTO outer_table values ( $1 )
    103 | 3345 | 23636 | UTILITY | COMMIT
```

L'esempio seguente mostra che il cursore `cur1` è stato chiuso quando l'istruzione TRUNCATE ha eseguito il commit.

```
CREATE OR REPLACE PROCEDURE sp_open_cursor_truncate()
LANGUAGE plpgsql
AS $$
DECLARE
  rec RECORD;
  cur1 cursor for select * from test_table_a order by 1;
BEGIN
  open cur1;
  TRUNCATE table test_table_b;
  Loop
    fetch cur1 into rec;
    raise info '%', rec.c1;
    exit when not found;
  End Loop;
END
$$;

call sp_open_cursor_truncate();
ERROR: cursor "cur1" does not exist
CONTEXT: PL/pgSQL function "sp_open_cursor_truncate" line 8 at fetch
```

L'esempio seguente emette un'istruzione TRUNCATE e non può essere chiamata da un blocco di transazioni esplicite.

```
CREATE OR REPLACE PROCEDURE sp_truncate_atomic() LANGUAGE plpgsql
AS $$
BEGIN
```

```
  TRUNCATE test_table_b;
END;
$$;

Begin;
  Call sp_truncate_atomic();
ERROR: TRUNCATE cannot be invoked from a procedure that is executing in an atomic
 context.
HINT: Try calling the procedure as a top-level call i.e. not from within an explicit
 transaction block.
Or, if this procedure (or one of its ancestors in the call chain) was created with SET
 config options, recreate the procedure without them.
CONTEXT: SQL statement "TRUNCATE test_table_b"
PL/pgSQL function "sp_truncate_atomic" line 2 at SQL statement
```

L'esempio seguente mostra che un utente che non è un utente con privilegi avanzati o un proprietario di una tabella può emettere un'istruzione TRUNCATE sulla tabella. L'utente lo fa usando una procedura archiviata `Security Definer`. Nell'esempio vengono illustrate le seguenti operazioni:

- user1 crea la tabella `test_tbl`.
- user1 crea la procedura archiviata `sp_truncate_test_tbl`.
- user1 concede il privilegio EXECUTE sulla procedura archiviata a user2.
- user2 esegue la procedura archiviata per troncare la tabella `test_tbl`. L'esempio mostra il conteggio delle righe prima e dopo il comando TRUNCATE.

```
set session_authorization to user1;
create table test_tbl(id int, name varchar(20));
insert into test_tbl values (1,'john'), (2, 'mary');
CREATE OR REPLACE PROCEDURE sp_truncate_test_tbl() LANGUAGE plpgsql
AS $$
DECLARE
  tbl_rows int;
BEGIN
  select count(*) into tbl_rows from test_tbl;
  RAISE INFO 'RowCount before Truncate: %', tbl_rows;
  TRUNCATE test_tbl;
  select count(*) into tbl_rows from test_tbl;
  RAISE INFO 'RowCount after Truncate: %', tbl_rows;
END;
$$ SECURITY DEFINER;
```

```
grant execute on procedure sp_truncate_test_tbl() to user2;
reset session_authorization;


set session_authorization to user2;
call sp_truncate_test_tbl();
INFO:  RowCount before Truncate: 2
INFO:  RowCount after Truncate: 0
CALL
reset session_authorization;
```

L'esempio seguente emette il COMMIT due volte. Il primo COMMIT esegue tutto il lavoro fatto nella transazione 10363 e avvia implicitamente la transazione 10364. La transazione 10364 viene eseguita dalla. seconda istruzione di COMMIT

```
CREATE OR REPLACE PROCEDURE sp_commit(a int, b int) LANGUAGE plpgsql
AS $$
BEGIN
  INSERT INTO test_table values (a);
  COMMIT;
  INSERT INTO test_table values (b);
  COMMIT;
END;
$$;

call sp_commit(1,2);

select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;
 userid |  xid  | pid  |  type   |
                              stmt_text
--------+-------+------+---------
+-----------------------------------------------------------------------------------------------
    100 | 10363 | 3089 | UTILITY | call sp_commit(1,2);
    100 | 10363 | 3089 | QUERY   | INSERT INTO test_table values ( $1 )
    100 | 10363 | 3089 | UTILITY | COMMIT
    100 | 10364 | 3089 | QUERY   | INSERT INTO test_table values ( $1 )
    100 | 10364 | 3089 | UTILITY | COMMIT
```

L'esempio seguente emette un'istruzione di ROLLBACK se `sum_vals` è maggiore di 2. La prima istruzione di ROLLBACK esegue il rollback di tutto il lavoro fatto nella transazione 10377 e avvia una nuova transazione 10378. La transazione 10378 viene confermata all'uscita della procedura.

```
CREATE OR REPLACE PROCEDURE sp_rollback(a int, b int) LANGUAGE plpgsql
AS $$
DECLARE
  sum_vals int;
BEGIN
  INSERT INTO test_table values (a);
  SELECT sum(c1) into sum_vals from test_table;
  IF sum_vals > 2 THEN
    ROLLBACK;
  END IF;

  INSERT INTO test_table values (b);
END;
$$;

call sp_rollback(1, 2);

select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;

userid |  xid  | pid  |  type   |
                              stmt_text
--------+-------+------+---------
+-----------------------------------------------------------------------------------------------
    100 | 10377 | 3089 | UTILITY | call sp_rollback(1, 2);
    100 | 10377 | 3089 | QUERY   | INSERT INTO test_table values ( $1 )
    100 | 10377 | 3089 | QUERY   | SELECT sum(c1) from test_table
    100 | 10377 | 3089 | QUERY   | Undoing 1 transactions on table 133646 with current
 xid 10377 : 10377
    100 | 10378 | 3089 | QUERY   | INSERT INTO test_table values ( $1 )
    100 | 10378 | 3089 | UTILITY | COMMIT
```

Gestione delle transazioni con procedura archiviata in modalità NONATOMIC

Una procedura archiviata creata in modalità NONATOMIC ha un comportamento di controllo delle transazioni diverso da una procedura creata in modalità predefinita. Analogamente al comportamento di commit automatico dei comandi SQL all'esterno delle procedure archiviate, ogni istruzione SQL

all'interno di una procedura NONATOMIC viene eseguita nella propria transazione e il commit è automatico. Se un utente inizia un blocco di transazione esplicito all'interno di una procedura archiviata NONATOMIC, le istruzioni SQL all'interno del blocco non effettuano il commit automatico. Il blocco delle transazioni controlla il commit o il rollback delle istruzioni al suo interno.

Nelle procedure archiviate NONATOMIC, è possibile aprire un blocco di transazioni esplicito all'interno della procedura utilizzando l'istruzione START TRANSACTION. Tuttavia, se esiste già un blocco di transazione aperto, questa istruzione non servirà a nulla perché Amazon Redshift non supporta le transazioni secondarie. La transazione precedente continua.

Quando si utilizzano i loop FOR del cursore all'interno di una procedura NONATOMIC, assicurati di aprire un blocco di transazioni esplicito prima di iterare i risultati di una query. Altrimenti, il cursore viene chiuso quando l'istruzione SQL all'interno del loop viene salvata automaticamente.

Alcune delle considerazioni relative all'utilizzo del comportamento in modalità NONATOMIC sono le seguenti:

- Ogni istruzione SQL all'interno della procedura archiviata viene sottoposta a commit automaticamente se non è presente un blocco di transazione aperto e la sessione ha il commit automatico impostato su ON.
- Puoi emettere un'istruzione COMMIT/ROLLBACK/TRUNCATE per terminare la transazione, se la procedura archiviata viene richiamata da un blocco di transazioni. Questo non è possibile nella modalità predefinita.
- Puoi emettere un'istruzione START TRANSACTION per iniziare un blocco di transazioni all'interno della procedura archiviata.

Gli esempi seguenti dimostrano il comportamento delle transazioni quando si utilizzano procedure archiviate NONATOMIC. La sessione per tutti i seguenti esempi ha il commit automatico impostato su ON.

Nell'esempio seguente, una procedura archiviata TRUNATOMIC ha due istruzioni INSERT. Quando la procedura viene richiamata all'esterno di un blocco di transazioni, ogni istruzione INSERT all'interno della procedura esegue automaticamente il commit.

```
CREATE TABLE test_table_a(v int);
CREATE TABLE test_table_b(v int);

CREATE OR REPLACE PROCEDURE sp_nonatomic_insert_table_a(a int, b int) NONATOMIC AS
$$
```

```
BEGIN
    INSERT INTO test_table_a values (a);
    INSERT INTO test_table_b values (b);
END;
$$
LANGUAGE plpgsql;

Call sp_nonatomic_insert_table_a(1,2);

Select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;

 userid | xid  |    pid     |  type   |               stmt_text
--------+------+------------+---------+----------------------------------------
      1 | 1792 | 1073807554 | UTILITY | Call sp_nonatomic_insert_table_a(1,2);
      1 | 1792 | 1073807554 | QUERY   | INSERT INTO test_table_a values ( $1 )
      1 | 1792 | 1073807554 | UTILITY | COMMIT
      1 | 1793 | 1073807554 | QUERY   | INSERT INTO test_table_b values ( $1 )
      1 | 1793 | 1073807554 | UTILITY | COMMIT
(5 rows)
```

Tuttavia, quando la procedura viene richiamata dall'interno di un blocco BEGIN..COMMIT, tutte le istruzioni fanno parte della stessa transazione (xid=1799).

```
Begin;
  INSERT INTO test_table_a values (10);
  Call sp_nonatomic_insert_table_a(20,30);
  INSERT INTO test_table_b values (40);
Commit;

Select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;

 userid | xid  |    pid     |  type   |                stmt_text
--------+------+------------+---------+-----------------------------------------
      1 | 1799 | 1073914035 | UTILITY | Begin;
      1 | 1799 | 1073914035 | QUERY   | INSERT INTO test_table_a values (10);
      1 | 1799 | 1073914035 | UTILITY | Call sp_nonatomic_insert_table_a(20,30);
      1 | 1799 | 1073914035 | QUERY   | INSERT INTO test_table_a values ( $1 )
      1 | 1799 | 1073914035 | QUERY   | INSERT INTO test_table_b values ( $1 )
      1 | 1799 | 1073914035 | QUERY   | INSERT INTO test_table_b values (40);
```

```
      1 | 1799 | 1073914035 | UTILITY | COMMIT
(7 rows)
```

In questo esempio, due istruzioni INSERT sono comprese tra START TRANSACTION...COMMIT. Quando la procedura viene richiamata all'esterno di un blocco di transazioni, le due istruzioni INSERT si trovano nella stessa transazione (xid=1866).

```
CREATE OR REPLACE PROCEDURE sp_nonatomic_txn_block(a int, b int) NONATOMIC AS
$$
BEGIN
    START TRANSACTION;
    INSERT INTO test_table_a values (a);
    INSERT INTO test_table_b values (b);
    COMMIT;
END;
$$
LANGUAGE plpgsql;

Call sp_nonatomic_txn_block(1,2);

Select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;

 userid | xid  |    pid     |  type   |               stmt_text
--------+------+------------+---------+----------------------------------------
      1 | 1865 | 1073823998 | UTILITY | Call sp_nonatomic_txn_block(1,2);
      1 | 1866 | 1073823998 | QUERY   | INSERT INTO test_table_a values ( $1 )
      1 | 1866 | 1073823998 | QUERY   | INSERT INTO test_table_b values ( $1 )
      1 | 1866 | 1073823998 | UTILITY | COMMIT
(4 rows)
```

Quando la procedura viene richiamata dall'interno di un blocco BEGIN...COMMIT, l'istruzione START TRANSACTION all'interno della procedura non esegue alcuna operazione perché esiste già una transazione aperta. L'istruzione COMMIT all'interno della procedura conferma la transazione corrente (xi=1876) e ne avvia una nuova.

```
Begin;
  INSERT INTO test_table_a values (10);
  Call sp_nonatomic_txn_block(20,30);
  INSERT INTO test_table_b values (40);
Commit;
```

```
Select userid, xid, pid, type, trim(text) as stmt_text
from svl_statementtext where pid = pg_backend_pid() order by xid , starttime ,
 sequence;

 userid | xid  |    pid     |  type   |               stmt_text
--------+------+------------+---------+----------------------------------------
      1 | 1876 | 1073832133 | UTILITY | Begin;
      1 | 1876 | 1073832133 | QUERY   | INSERT INTO test_table_a values (10);
      1 | 1876 | 1073832133 | UTILITY | Call sp_nonatomic_txn_block(20,30);
      1 | 1876 | 1073832133 | QUERY   | INSERT INTO test_table_a values ( $1 )
      1 | 1876 | 1073832133 | QUERY   | INSERT INTO test_table_b values ( $1 )
      1 | 1876 | 1073832133 | UTILITY | COMMIT
      1 | 1878 | 1073832133 | QUERY   | INSERT INTO test_table_b values (40);
      1 | 1878 | 1073832133 | UTILITY | COMMIT
(8 rows)
```

Questo esempio illustra come utilizzare i loop del cursore. La tabella test_table_a ha tre valori. L'obiettivo è effettuare l'iterazione dei tre valori e inserirli nella tabella test_table_b. Se una procedura archiviata NONATOMIC viene creata nel modo seguente, genererà l'errore cursor "cur1" does not exist (cursore "cur1" inesistente) dopo l'esecuzione dell'istruzione INSERT nel primo ciclo. Questo perché il commit automatico di INSERT chiude il cursore aperto.

```
insert into test_table_a values (1), (2), (3);

CREATE OR REPLACE PROCEDURE sp_nonatomic_cursor() NONATOMIC
LANGUAGE plpgsql
AS $$
DECLARE
  rec RECORD;
  cur1 cursor for select * from test_table_a order by 1;
BEGIN
  open cur1;
  Loop
    fetch cur1 into rec;
    exit when not found;
    raise info '%', rec.v;
    insert into test_table_b values (rec.v);
  End Loop;
END
$$;

CALL sp_nonatomic_cursor();
```

```
INFO:  1
ERROR:  cursor "cur1" does not exist
CONTEXT:  PL/pgSQL function "sp_nonatomic_cursor" line 7 at fetch
```

Per far funzionare il loop del cursore, inserirlo tra START TRANSACTION...COMMIT.

```
insert into test_table_a values (1), (2), (3);

CREATE OR REPLACE PROCEDURE sp_nonatomic_cursor() NONATOMIC
LANGUAGE plpgsql
AS $$
DECLARE
  rec RECORD;
  cur1 cursor for select * from test_table_a order by 1;
BEGIN
  START TRANSACTION;
  open cur1;
  Loop
    fetch cur1 into rec;
    exit when not found;
    raise info '%', rec.v;
    insert into test_table_b values (rec.v);
  End Loop;
  COMMIT;
END
$$;

CALL sp_nonatomic_cursor();

INFO:  1
INFO:  2
INFO:  3
CALL
```

Errori di blocco

Quando una query o un comando in una procedura archiviata causa un errore, le query successive non vengono eseguite e viene eseguito il rollback della transazione. Però puoi gestire gli errori usando un blocco EXCEPTION.

> **Note**
>
> Il comportamento predefinito prevede che un errore impedisca l'esecuzione delle query successive, anche quando non sono presenti ulteriori condizioni che generano errori nella procedura archiviata.

```
[ <<label>> ]
[ DECLARE
  declarations ]
BEGIN
  statements
EXCEPTION
  WHEN OTHERS THEN
    statements
END;
```

Quando si verifica un'eccezione e si aggiunge un blocco di gestione delle eccezioni, è possibile scrivere istruzioni RAISE e la maggior parte delle altre istruzioni PL/pgSQL. Ad esempio, è possibile generare un'eccezione con un messaggio personalizzato o inserire un record in una tabella di registrazione.

Quando si inserisce il blocco di gestione delle eccezioni, la transazione corrente viene ripristinata e viene creata una nuova transazione per eseguire le istruzioni nel blocco. Se le istruzioni nel blocco vengono eseguite senza errori, la transazione viene eseguita e l'eccezione viene rigenerata. Infine, la procedura archiviata termina.

L'unica condizione supportata in un blocco dell’eccezione è OTHERS, che fa corrispondere qualsiasi tipo di errore ad eccezione dell'annullamento della query. Inoltre, se si verifica un errore in un blocco di gestione delle eccezioni, può essere preso da un blocco di gestione delle eccezioni esterno.

Quando si verifica un errore all'interno della procedura NONATOMIC, l'errore non viene generato nuovamente se viene gestito da un blocco di eccezioni. Consulta l'istruzione PL/pgSQL `RAISE` per generare nuovamente l'eccezione rilevata da un blocco di gestione delle eccezioni. Questa istruzione è valida solo nei blocchi di gestione delle eccezioni. Per ulteriori informazioni, consulta RAISE.

Controllo di ciò che accade dopo un errore in una stored procedure con il gestore CONTINUE

Il gestore `CONTINUE` è un tipo di gestore di eccezioni che controlla il flusso di esecuzione in una stored procedure NONATOMIC. Ti consente di acquisire e gestire le eccezioni senza terminare il

blocco di istruzioni esistente. In genere, quando si verifica un errore in una stored procedure, il flusso viene interrotto e l'errore viene restituito al chiamante. Tuttavia, in alcuni casi d'uso, la condizione di errore non è sufficientemente grave da giustificare l'interruzione del flusso. Potresti voler gestire l'errore in modo corretto, utilizzando una logica di gestione degli errori di tua scelta in una transazione separata, e quindi continuare a eseguire le istruzioni dopo l'errore. L'esempio seguente mostra la sintassi.

```
[ DECLARE
  declarations ]
BEGIN
  statements
EXCEPTION
  [ CONTINUE_HANDLER | EXIT_HANDLER ] WHEN OTHERS THEN
    handler_statements
END;
```

Sono disponibili diverse tabelle di sistema che consentono di raccogliere informazioni su vari tipi di errore. Per ulteriori informazioni, consulta STL_LOAD_ERRORS, STL_ERROR e SYS_STREAM_SCAN_ERRORS. Esistono anche ulteriori tabelle di sistema che è possibile utilizzare per risolvere gli errori. Ulteriori informazioni sono reperibili in Riferimento di tabelle e viste di sistema.

Esempio

Nell'esempio seguente viene illustrato come scrivere istruzioni nel blocco di gestione delle eccezioni. La procedura archiviata utilizza il comportamento predefinito di gestione delle transazioni.

```
CREATE TABLE employee (firstname varchar, lastname varchar);
INSERT INTO employee VALUES ('Tomas','Smith');
CREATE TABLE employee_error_log (message varchar);

CREATE OR REPLACE PROCEDURE update_employee_sp() AS
$$
BEGIN
    UPDATE employee SET firstname = 'Adam' WHERE lastname = 'Smith';
    EXECUTE 'select invalid';
EXCEPTION WHEN OTHERS THEN
    RAISE INFO 'An exception occurred.';
    INSERT INTO employee_error_log VALUES ('Error message: ' || SQLERRM);
END;
$$
LANGUAGE plpgsql;
```

```
CALL update_employee_sp();

INFO:  An exception occurred.
ERROR:  column "invalid" does not exist
CONTEXT:  SQL statement "select invalid"
PL/pgSQL function "update_employee_sp" line 3 at execute statement
```

In questo esempio, viene chiamato `update_employee_sp`, il messaggio informativo Si è verificata un'eccezione. viene sollevato e il messaggio di errore viene inserito nel log `employee_error_log` della tabella di registrazione. L'eccezione originale viene generata nuovamente prima dell'uscita della procedura archiviata. Le query seguenti mostrano i record derivanti dall'esecuzione dell'esempio.

```
SELECT * from employee;

firstname | lastname
-----------+----------
 Tomas     | Smith

SELECT * from employee_error_log;

          message
-----------------------------------------------
 Error message: column "invalid" does not exist
```

Per ulteriori informazioni su RAISE, inclusi la guida alla formattazione e un elenco di ulteriori livelli, consulta Istruzioni PL/pgSQL supportate.

Nell'esempio seguente viene illustrato come scrivere istruzioni nel blocco di gestione delle eccezioni. La procedura archiviata utilizza il comportamento NONATOMIC di gestione delle transazioni. In questo esempio, non viene restituito alcun errore al chiamante dopo il completamento della chiamata di procedura. Non viene eseguito il roll back dell'istruzione UPDATE a causa dell'errore nell'istruzione successiva. Il messaggio informativo viene generato e il messaggio di errore viene inserito nella tabella di logging.

```
CREATE TABLE employee (firstname varchar, lastname varchar);
INSERT INTO employee VALUES ('Tomas','Smith');
CREATE TABLE employee_error_log (message varchar);

-- Create the SP in NONATOMIC mode
CREATE OR REPLACE PROCEDURE update_employee_sp_2() NONATOMIC AS
```

```
$$
BEGIN
    UPDATE employee SET firstname = 'Adam' WHERE lastname = 'Smith';
    EXECUTE 'select invalid';
EXCEPTION WHEN OTHERS THEN
    RAISE INFO 'An exception occurred.';
    INSERT INTO employee_error_log VALUES ('Error message: ' || SQLERRM);
END;
$$
LANGUAGE plpgsql;

CALL update_employee_sp_2();
INFO:  An exception occurred.
CALL

SELECT * from employee;

 firstname | lastname
-----------+----------
 Adam      | Smith
(1 row)

SELECT * from employee_error_log;

                    message
-----------------------------------------------
 Error message: column "invalid" does not exist
(1 row)
```

Questo esempio illustra come creare una procedura con due blocchi secondari. Quando viene richiamata la procedura archiviata, l'errore del primo blocco secondario viene gestito dal relativo blocco di gestione delle eccezioni. Una volta completato il primo blocco secondario, la procedura continua a eseguire il secondo blocco secondario. È possibile vedere dal risultato che non viene generato alcun errore al termine della chiamata di procedura. Viene eseguito il commit delle operazioni UPDATE e INSERT sulla tabella employee. I messaggi di errore di entrambi i blocchi di eccezioni vengono inseriti nella tabella di logging.

```
CREATE TABLE employee (firstname varchar, lastname varchar);
INSERT INTO employee VALUES ('Tomas','Smith');
CREATE TABLE employee_error_log (message varchar);

CREATE OR REPLACE PROCEDURE update_employee_sp_3() NONATOMIC AS
```

```
$$
BEGIN
    BEGIN
        UPDATE employee SET firstname = 'Adam' WHERE lastname = 'Smith';
        EXECUTE 'select invalid1';
    EXCEPTION WHEN OTHERS THEN
        RAISE INFO 'An exception occurred in the first block.';
        INSERT INTO employee_error_log VALUES ('Error message: ' || SQLERRM);
    END;
    BEGIN
        INSERT INTO employee VALUES ('Edie','Robertson');
        EXECUTE 'select invalid2';
    EXCEPTION WHEN OTHERS THEN
        RAISE INFO 'An exception occurred in the second block.';
        INSERT INTO employee_error_log VALUES ('Error message: ' || SQLERRM);
    END;
END;
$$
LANGUAGE plpgsql;

CALL update_employee_sp_3();
INFO:  An exception occurred in the first block.
INFO:  An exception occurred in the second block.
CALL

SELECT * from employee;

 firstname | lastname
-----------+-----------
 Adam      | Smith
 Edie      | Robertson
(2 rows)

SELECT * from employee_error_log;

                     message
-----------------------------------------------
 Error message: column "invalid1" does not exist
 Error message: column "invalid2" does not exist
(2 rows)
```

Nell'esempio seguente viene mostrato come utilizzare il gestore di eccezioni CONTINUE. Questo esempio crea due tabelle e le utilizza in una stored procedure. Il gestore CONTINUE controlla il

flusso di esecuzione in una stored procedure con un comportamento di gestione delle transazioni NONATOMIC.

```
CREATE TABLE tbl_1 (a int);
CREATE TABLE tbl_error_logging(info varchar, err_state varchar, err_msg varchar);

CREATE OR REPLACE PROCEDURE sp_exc_handling_1() NONATOMIC AS
$$
BEGIN
    INSERT INTO tbl_1 VALUES (1);
    -- Expect an error for the insert statement following, because of the invalid value
    INSERT INTO tbl_1 VALUES ("val");
    INSERT INTO tbl_1 VALUES (2);
EXCEPTION CONTINUE_HANDLER WHEN OTHERS THEN
    INSERT INTO tbl_error_logging VALUES ('Encountered error', SQLSTATE, SQLERRM);
END;
$$ LANGUAGE plpgsql;
```

Chiama la stored procedure:

```
CALL sp_exc_handling_1();
```

Il flusso procede in questo modo:

1. Si verifica un errore perché si tenta di inserire in una colonna un tipo di dati non compatibile. Il controllo passa al blocco EXCEPTION. Quando si inserisce il blocco di gestione delle eccezioni, viene eseguito il rollback della transazione corrente e viene creata una nuova transazione implicita per eseguire le istruzioni nel blocco.
2. Se le istruzioni in CONTINUE_HANDLER vengono eseguite senza errori, il controllo passa all'istruzione immediatamente successiva a quella che ha causato l'eccezione. Se un'istruzione in CONTINUE_HANDLER genera una nuova eccezione, puoi gestirla con un gestore di eccezioni all'interno del blocco EXCEPTION.

Dopo aver chiamato la stored procedure di esempio, le tabelle contengono i seguenti record:

- Se esegui `SELECT * FROM tbl_1;`, vengono restituiti due record che contengono i valori 1 e 2.
- Se esegui `SELECT * FROM tbl_error_logging;`, viene restituito un record con questi valori: Encountered error, 42703 e column "val" does not exist in tbl_1.

Il seguente esempio di gestione degli errori utilizza sia un gestore EXIT sia un gestore CONTINUE. Crea due tabelle: una tabella di dati e una tabella di log. Crea inoltre una stored procedure per la gestione degli errori:

```
CREATE TABLE tbl_1 (a int);
CREATE TABLE tbl_error_logging(info varchar, err_state varchar, err_msg varchar);

CREATE OR REPLACE PROCEDURE sp_exc_handling_2() NONATOMIC AS
$$
BEGIN
    INSERT INTO tbl_1 VALUES (1);
    BEGIN
        INSERT INTO tbl_1 VALUES (100);
        -- Expect an error for the insert statement following, because of the invalid
 value
        INSERT INTO tbl_1 VALUES ("val");
        INSERT INTO tbl_1 VALUES (101);
    EXCEPTION EXIT_HANDLER WHEN OTHERS THEN
        INSERT INTO tbl_error_logging VALUES ('Encountered error', SQLSTATE, SQLERRM);
    END;
    INSERT INTO tbl_1 VALUES (2);
    -- Expect an error for the insert statement following, because of the invalid value
    INSERT INTO tbl_1 VALUES ("val");
    INSERT INTO tbl_1 VALUES (3);
EXCEPTION CONTINUE_HANDLER WHEN OTHERS THEN
    INSERT INTO tbl_error_logging VALUES ('Encountered error', SQLSTATE, SQLERRM);
END;
$$ LANGUAGE plpgsql;
```

Dopo aver creato la stored procedure, chiamala usando la seguente istruzione:

```
CALL sp_exc_handling_2();
```

Quando si verifica un errore nel blocco di eccezioni interno, che è racchiuso tra parentesi dal set interno di BEGIN e END, viene gestito dal gestore EXIT. Tutti gli errori che si verificano nel blocco esterno vengono gestiti dal gestore CONTINUE.

Dopo aver chiamato la stored procedure di esempio, le tabelle contengono i seguenti record:

- Se esegui `SELECT * FROM tbl_1;`, vengono restituiti quattro record con i valori 1, 2, 3 e 100.

- Se esegui `SELECT * FROM tbl_error_logging;`, vengono restituiti due record con questi valori: Encountered error, 42703 e column "val" does not exist in tbl_1.

Se la tabella tbl_error_logging non esiste viene generata un'eccezione.

Nell'esempio seguente viene mostrato come utilizzare il gestore di eccezioni CONTINUE con il ciclo FOR. Questo esempio crea tre tabelle e le utilizza in un ciclo FOR all'interno di una stored procedure. Il ciclo FOR è una variante del set di risultati, il che significa che esegue l'iterazione dei risultati di una query:

```
CREATE TABLE tbl_1 (a int);
INSERT INTO tbl_1 VALUES (1), (2), (3);
CREATE TABLE tbl_2 (a int);
CREATE TABLE tbl_error_logging(info varchar, err_state varchar, err_msg varchar);

CREATE OR REPLACE PROCEDURE sp_exc_handling_loop() NONATOMIC AS
$$
DECLARE
 rec RECORD;
BEGIN
    FOR rec IN SELECT a FROM tbl_1
    LOOP
        IF rec.a = 2 THEN
            -- Expect an error for the insert statement following, because of the
 invalid value
            INSERT INTO tbl_2 VALUES("val");
        ELSE
            INSERT INTO tbl_2 VALUES (rec.a);
        END IF;
    END LOOP;
EXCEPTION CONTINUE_HANDLER WHEN OTHERS THEN
    INSERT INTO tbl_error_logging VALUES ('Encountered error', SQLSTATE, SQLERRM);
END;
$$ LANGUAGE plpgsql;
```

Chiama la stored procedure:

```
CALL sp_exc_handling_loop();
```

Dopo aver chiamato la stored procedure di esempio, le tabelle contengono i seguenti record:

- Se esegui `SELECT * FROM tbl_2;`, vengono restituiti due record che contengono i valori 1 e 3.
- Se esegui `SELECT * FROM tbl_error_logging;`, viene restituito un record con questi valori: Encountered error, 42703 e column "val" does not exist in tbl_2.

Note per l'utilizzo relative all'handler CONTINUE:

- Le parole chiave CONTINUE_HANDLER ed EXIT_HANDLER possono essere utilizzate solo nelle stored procedure NONATOMIC.
- Le parole chiave CONTINUE_HANDLER ed EXIT_HANDLER sono facoltative. EXIT_HANDLER è l'impostazione predefinita.

Registrazione delle stored procedure

I dettagli sulle procedure archiviate vengono registrate nelle tabelle e visualizzazioni di sistema seguenti:

- SVL_STORED_PROC_CALL: vengono registrati i dettagli sull'ora di inizio e di fine della chiamata della procedura archiviata e riporta se la chiamata viene terminata prima del completamento. Per ulteriori informazioni, consultare SVL_STORED_PROC_CALL.
- SVL_STORED_PROC_MESSAGES: i messaggi nelle procedure archiviate emessi dalla query RAISE vengono registrati con il livello di registrazione corrispondente. Per ulteriori informazioni, consultare SVL_STORED_PROC_MESSAGES.
- SVL_QLOG: l'ID query della chiamata della procedura archiviata viene registrato per ogni query chiamata da una procedura archiviata. Per ulteriori informazioni, consultare SVL_QLOG.
- STL_UTILITYTEXT: le chiamate alle procedure archiviate vengono registrate dopo il loro completamento. Per ulteriori informazioni, consultare STL_UTILITYTEXT.
- PG_PROC_INFO: questa vista del catalogo di sistema mostra informazioni sulle procedure archiviate. Per ulteriori informazioni, consulta PG_PROC_INFO.

Considerazioni per il supporto delle procedure archiviate

Le seguenti considerazioni si applicano quando si utilizzano le procedure archiviate in Amazon Redshift.

Differenze tra Amazon Redshift e PostgreSQL per il supporto alle procedure archiviate

Di seguito sono riportate le differenze tra il supporto delle procedure archiviate in Amazon Redshift e in PostgreSQL:

- Amazon Redshift non supporta le transazioni secondarie e quindi ha supporto limitato per i blocchi di gestione delle eccezioni.

Considerazioni e limiti

Di seguito sono riportate le considerazioni sulle procedure archiviate in Amazon Redshift:

- Il numero massimo di procedure archiviate per un database è 10.000.
- Le dimensioni massime del codice di origine per una procedura è 2 MB.
- Il numero massimo di cursori espliciti e impliciti che puoi aprire simultaneamente in una sessione utente è uno. I loop FOR che eseguono le iterazioni sul set di risultati di un'istruzione SQL aprono cursori impliciti. I cursori nidificati non sono supportati.
- I cursori espliciti e impliciti hanno le stesse limitazioni sulle dimensioni del set di risultati dei cursori Amazon Redshift standard. Per ulteriori informazioni, consultare Vincoli del cursore.
- Il numero massimo di livelli per le chiamate nidificate è 16.
- Il numero massimo di parametri di procedura è 32 per gli argomenti di input e 32 per gli argomenti di output.
- Il numero massimo di variabili in una procedura archiviata è 1024.
- Qualsiasi comando SQL che richiede il proprio contesto di transazione non è supportato in una procedura archiviata. Esempi includono:
  - PREPARE
  - CREATE/DROP DATABASE
  - CREATE EXTERNAL TABLE
  - VACUUM
  - SET LOCALE
  - ALTER TABLE APPEND
- La chiamata di metodo `registerOutParameter` tramite il driver Java Database Connectivity (JDBC) non è supportato per il tipo di dati `refcursor`. Per un esempio di utilizzo del tipo di dati `refcursor`, consultare Restituzione di un set di risultati.

Riferimento al linguaggio PL/pgSQL

Le procedure archiviate in Amazon Redshift si basano sul linguaggio procedurale PL/pgSQL di PostgreSQL, con alcune differenze importanti. In questo riferimento, è possibile trovare dettagli della sintassi PL/pgSQL così come implementata da Amazon Redshift. Per ulteriori informazioni su PL/pgSQL, consultare Linguaggio procedurale SQL PL/pgSQL nella documentazione di PostgreSQL.

Argomenti

Convenzioni del riferimento PL/pgSQL

In questa sezione, puoi trovare le convenzioni utilizzate per scrivere la sintassi per il linguaggio procedurale archiviato PL/pgSQL.

Carattere	Descrizione
CAPS	Le parole in lettere maiuscole sono parole chiave.
[ ]	Le parentesi indicano argomenti opzionali. Più argomenti tra parentesi indicano che è possibile scegliere qualsiasi numero degli argomenti. Inoltre, gli argomenti tra parentesi su righe separate indicano che il parser di Amazon Redshift prevede che gli argomenti siano nell'ordine in cui sono elencati nella sintassi.
{ }	Le parentesi graffe indicano che è necessario scegliere uno degli argomenti racchiusi nelle stesse.
|	Le pipe indicano che è possibile scegliere tra gli argomenti.
corsivo rosso	Le parole in corsivo rosso indicano dei segnaposto. Inserisci il valore appropriato al posto della parola in corsivo.
. . .	I puntini di sospensione indicano che è possibile ripetere l'elemento precedente.

Carattere	Descrizione
'	Le parole tra virgolette singole indicano che è necessario digitare le virgolett e.

Struttura di PL/pgSQL

PL/pgSQL è un linguaggio procedurale con molte delle stesse costruzioni di altri linguaggi procedurali.

Argomenti

- Blocco
- Dichiarazione di variabile
- Dichiarazione alias
- Variabili integrate
- Tipi di record

Blocco

PL/pgSQL è un linguaggio strutturato a blocchi. Il corpo completo di una procedura è definito in un blocco, che contiene dichiarazioni variabili e istruzioni PL/pgSQL. Un'istruzione può anche essere un blocco nidificato o un sottoblocco.

Finire le dichiarazioni e le istruzioni con un punto e virgola. Segui la parola chiave END in un blocco o un sottoblocco con un punto e virgola. Non utilizzare il punto e virgola dopo le parole chiave DECLARE e BEGIN.

Puoi scrivere tutte le parole chiave e tutti gli identificatori in un mix di maiuscolo e minuscolo. Gli identificatori vengono convertiti implicitamente in minuscolo a meno che siano racchiusi in virgolette doppie.

Un trattino doppio (--) inizia un commento che si estende alla fine della riga. Un /* inizia un commento blocco che si estende alla prossima occorrenza di */. Non è possibile nidificare commenti blocco. Tuttavia, è possibile racchiudere i commenti a doppio trattino in un commento blocco e un trattino doppio può nascondere i delimitatori del commento blocco /* e */.

Qualsiasi istruzione in una sezione istruzioni di un blocco può essere un sottoblocco. Puoi utilizzare sottoblocchi per un raggruppamento logico o per localizzare variabili di un gruppo piccolo di istruzioni.

```
[ <<label>> ]
[ DECLARE
  declarations ]
BEGIN
  statements
END [ label ];
```

Le variabili dichiarate nella sezione dichiarazioni prima di un blocco vengono inizializzate ai loro valori predefiniti ogni volta che viene inserito il blocco. In altre parole, non vengono inizializzati solo una volta per chiamata di funzione.

Di seguito viene riportato un esempio.

```
CREATE PROCEDURE update_value() AS $$
DECLARE
  value integer := 20;
BEGIN
  RAISE NOTICE 'Value here is %', value;  -- Value here is 20
  value := 50;
  --
  -- Create a subblock
  --
  DECLARE
    value integer := 80;
  BEGIN
    RAISE NOTICE 'Value here is %', value;  -- Value here is 80
  END;

  RAISE NOTICE 'Value here is %', value;  -- Value here is 50
END;
$$ LANGUAGE plpgsql;
```

Utilizza un'etichetta per identificare il blocco da utilizzare in un'istruzione EXIT o per qualificare i nome delle variabili dichiarate nel blocco.

Non confondere l'utilizzo di BEGIN/END per le istruzioni di raggruppamento in PL/pgSQL con i comandi del database per il controllo della transazione. BEGIN e END in PL/pgSQL sono solo per il raggruppamento. Non iniziano o terminano la transazione.

Dichiarazione di variabile

Dichiarare tutte le variabili in un blocco, ad eccezione delle variabili di loop, nella sezione DECLARE del blocco. Le variabili possono utilizzare qualsiasi tipo di dati Amazon Redshift valido. Per tipi di dati supportati, consultare Tipi di dati.

Le variabili PL/pgSQL possono essere qualsiasi tipo di dati supportati da Amazon Redshift, più `RECORD` e `refcursor`. Per ulteriori informazioni su `RECORD`, consultare Tipi di record. Per ulteriori informazioni su `refcursor`, consultare Cursori.

```
DECLARE
name [ CONSTANT ] type [ NOT NULL ] [ { DEFAULT | := } expression ];
```

Seguono delle dichiarazioni delle variabili di esempio.

```
customerID integer;
numberofitems numeric(6);
link varchar;
onerow RECORD;
```

La variabile per un loop FOR che effettua l'iterazione in una gamma di numeri interi viene dichiarata automaticamente come variabile di numero intero.

La clausola DEFAULT, se fornita, specifica il valore iniziale assegnato alla variabile quando viene inserito il blocco. Se la clausola DEFAULT non viene fornita, la variabile viene inizializzata con il valore SQL NULL. L'opzione CONSTANT previene che la variabile venga assegnata, in modo che il suo valore rimanga constante per la durata del blocco. Se NOT NULL è specificato, un'assegnazione di un valore null risulta in errore di runtime. Tutte le variabili dichiarate come NOT NULL devono avere un valore predefinito non-null specificato.

Il valore predefinito viene valutato ogni volta che il blocco viene inserito. Ad esempio, l'assegnazione di `now()` a una variabile del tipo `timestamp` determina che la variabile abbia l'ora della chiamata di funzione corrente, non l'ora di quando la funzione è stata precompilata.

```
quantity INTEGER DEFAULT 32;
url VARCHAR := 'http://mysite.com';
user_id CONSTANT INTEGER := 10;
```

Il tipo di dati `refcursor` è il tipo di dati delle variabili di cursore nelle procedure archiviate. Un valore `refcursor` può essere restituito da una procedura archiviata. Per ulteriori informazioni, consultare Restituzione di un set di risultati.

Dichiarazione alias

Se la firma della procedura archiviata omette il nome dell'argomento, è possibile dichiarare un alias per l'argomento.

```
name ALIAS FOR $n;
```

Variabili integrate

Le seguenti variabili integrate sono supportate:

- FOUND
- SQLSTATE
- SQLERRM
- GET DIAGNOSTICS integer_var := ROW_COUNT;

FOUND è una variabile speciale di tipo booleano. FOUND inizia come falsa in ogni chiamata di procedura. FOUND viene impostato dai seguenti tipi di istruzioni:

- SELECT INTO

  Imposta FOUND su true se restituisce una riga, false se non viene restituita alcuna riga.
- UPDATE, INSERT e DELETE

  Imposta FOUND su true se almeno una riga è coinvolta, false se nessuna riga è coinvolta.
- FETCH

  Imposta FOUND su true se restituisce una riga, false se non viene restituita alcuna riga.
- Istruzione FOR

  Imposta FOUND su true se l'istruzione FOR effettua l'iterazione una o più volte, altrimenti la imposta su false. Questo si applica a tutte e tre le varianti dell'istruzione FOR: loop FOR interi, loop FOR record-set e loop FOR record-set dinamici.

FOUND viene impostato quando il loop FOR esce. Nel runtime del loop, FOUND non viene modificato dall'istruzione FOR. Tuttavia, può essere modificato dall'esecuzione di altre istruzioni nel corpo del loop.

Di seguito viene riportato un esempio.

```
CREATE TABLE employee(empname varchar);
CREATE OR REPLACE PROCEDURE show_found()
AS  $$
DECLARE
  myrec record;
BEGIN
  SELECT INTO myrec * FROM employee WHERE empname = 'John';
  IF NOT FOUND THEN
    RAISE EXCEPTION 'employee John not found';
  END IF;
END;
$$ LANGUAGE plpgsql;
```

Nel gestore delle eccezioni, la variabile speciale SQLSTATE contiene il codice di errore che corrisponde all'eccezione che è stata generata. La variabile speciale SQLERRM contiene il messaggio di errore associato all'eccezione. Queste variabili non sono definite al di fuori dei gestori delle eccezioni e, se utilizzate, generano un errore.

Di seguito viene riportato un esempio.

```
CREATE OR REPLACE PROCEDURE sqlstate_sqlerrm() AS
$$
BEGIN
  UPDATE employee SET firstname = 'Adam' WHERE lastname = 'Smith';
  EXECUTE 'select invalid';
  EXCEPTION WHEN OTHERS THEN
  RAISE INFO 'error message SQLERRM %', SQLERRM;
  RAISE INFO 'error message SQLSTATE %', SQLSTATE;
END;
$$ LANGUAGE plpgsql;
```

ROW_COUNT viene utilizzato con il comando GET DIAGNOSTICS. Mostra il numero di righe elaborate dall'ultimo comando SQL inviato al motore SQL.

Di seguito viene riportato un esempio.

```
CREATE OR REPLACE PROCEDURE sp_row_count() AS
$$
DECLARE
  integer_var int;
BEGIN
  INSERT INTO tbl_row_count VALUES(1);
  GET DIAGNOSTICS integer_var := ROW_COUNT;
  RAISE INFO 'rows inserted = %', integer_var;
END;
$$ LANGUAGE plpgsql;
```

Tipi di record

Un tipo RECORD non è un tipo di dati true, solo un segnaposto. Le variabili tipo di record assumono la struttura della riga attuale della riga alla quale sono assegnate durante un comando SELECT o FOR. La sottostruttura di una variabile di record può cambiare ogni volta che le viene assegnato un valore. Fino a quando una variabile di record non viene assegnata, non ha sottostruttura. Ogni tentativo di accedere un campo in essa genera un errore di runtime.

```
name RECORD;
```

Di seguito viene riportato un esempio.

```
CREATE TABLE tbl_record(a int, b int);
INSERT INTO tbl_record VALUES(1, 2);
CREATE OR REPLACE PROCEDURE record_example()
LANGUAGE plpgsql
AS $$
DECLARE
  rec RECORD;
BEGIN
  FOR rec IN SELECT a FROM tbl_record
  LOOP
    RAISE INFO 'a = %', rec.a;
  END LOOP;
END;
$$;
```

Istruzioni PL/pgSQL supportate

Le istruzioni PL/pgSQL aumentano i comandi SQL con costrutti procedurali, includendo le espressioni di looping e condizionali, per controllare il flusso logico. È possibile utilizzare la maggior parte dei comandi SQL, compresi il linguaggio DML (Data Manipulation Language) come COPY, UNLOAD e INSERT, e il linguaggio DDL (Data Definition Language) come CREATE TABLE. Per un elenco di comandi SQL esaustivi, consultare Comandi SQL. Inoltre, le seguenti istruzioni PL/pgSQL sono supportate da Amazon Redshift.

Argomenti

Assegnazione

L'istruzione di assegnazione assegna un valore a una variabile. L'espressione deve restituire un valore singolo.

```
identifier := expression;
```

L'utilizzo di = non standard per l'assegnazione, invece di `:=` è anche accettabile.

Se il tipo di dati dell'espressione non corrisponde al tipo di dati della variabile o la variabile ha una dimensione o precisione, il valore del risultato viene convertito implicitamente.

Di seguito vengono riportati degli esempi.

```
customer_number := 20;
tip := subtotal * 0.15;
```

SELECT INTO

L'istruzione SELECT INTO assegna il risultato di colonne multiple (ma solo una riga) in una variabile di record o in un elenco di variabili scalari.

```
SELECT INTO target select_expressions FROM ...;
```

Nella sintassi precedente, *target* può essere una variabile di record o un elenco separato da virgole di variabili semplici e campi record. L'elenco *select_expressions* e il resto del comando coincidono come nel SQL regolare.

Se un elenco di variabili viene utilizzato come *target*, i valori selezionati devono corrispondere esattamente alla struttura del target o si verifica un errore di runtime. Quando una variabile di record è il target, si configura automaticamente al tipo di riga delle colonne dei risultati delle query.

La clausola INTO può apparire quasi ovunque nell'istruzione SELECT. Solitamente appare appena dopo la clausola SELECT o appena prima della clausola FROM. Ovvero, appare appena prima o appena dopo l'elenco *select_expressions*.

Se la query restituisce zero righe, i valori NULL vengono assegnati al *target*. Se la query restituisce righe multiple, la prima riga viene assegnata al *target* e le altre vengono eliminate. A meno che l'istruzione contenga un ORDER BY, la prima riga non è deterministica.

Per determinare se l'assegnazione ha restituito almeno una riga, utilizza la variabile speciale FOUND.

```
SELECT INTO customer_rec * FROM cust WHERE custname = lname;
IF NOT FOUND THEN
  RAISE EXCEPTION 'employee % not found', lname;
END IF;
```

Per testare se un risultato di un record è null, è possibile utilizzare il condizionale IS NULL. Non c'è modo di determinare se delle righe aggiuntive sono state eliminate. L'esempio seguente gestisce il caso nel quale nessuna riga è stata restituita.

```
CREATE OR REPLACE PROCEDURE select_into_null(return_webpage OUT varchar(256))
AS $$
```

```
DECLARE
  customer_rec RECORD;
BEGIN
  SELECT INTO customer_rec * FROM users WHERE user_id=3;
  IF customer_rec.webpage IS NULL THEN
    -- user entered no webpage, return "http://"
    return_webpage = 'http://';
  END IF;
END;
$$ LANGUAGE plpgsql;
```

No-op

L'istruzione no-op (`NULL;`) è un'istruzione segnaposto che non fa nulla. Un'istruzione no-op può indicare un ramo di una catena IF-THEN-ELSE è vuoto.

```
NULL;
```

SQL dinamico

Per generare i comandi dinamici che possono includere diverse tabelle o diversi tipi di dati ogni volta che vengono eseguiti da una procedura archiviata PL/pgSQL, utilizza l'istruzione `EXECUTE`.

```
EXECUTE command-string [ INTO target ];
```

Nell'esempio precedente, *`command-string`* è un'espressione che produce una stringa (di tipo testo) che contiene il comando da eseguire. Questo valore *`command-string`* viene inviato al motore SQL. Nessuna sostituzione delle variabili PL/pgSQL viene effettuata sulla stringa del comando. I valori delle variabili devono essere inseriti nella stringa comando com'è costruito.

> **Note**
>
> Non è possibile utilizzare le istruzioni COMMIT e ROLLBACK dall'interno di un SQL dinamico. Per informazioni su come utilizzare le istruzioni COMMIT e ROLLBACK all'interno di una procedura archiviata, consultare Gestione delle transazioni.

Quando si lavora con i comandi dinamici, spesso è necessario necessario gestire l'escape di virgolette singole. Consigliamo di racchiudere il testo fisso in virgolette nel testo della funzione

utilizzando il dollar quoting. I valori dinamici da inserire in una query costruita richiedono una gestione speciale perché loro stessi potrebbero contenere virgolette. L'esempio seguente utilizza il dollar quoting per tutta la funzione, quindi le virgolette non devono essere raddoppiate.

```
EXECUTE 'UPDATE tbl SET '
  || quote_ident(colname)
  || ' = '
  || quote_literal(newvalue)
  || ' WHERE key = '
  || quote_literal(keyvalue);
```

L'esempio precedente mostra le funzioni `quote_ident(text)` e `quote_literal(text)`. Questo esempio passa variabili che contengono identificatori di colonna e tabella alla funzione `quote_ident`. Passa anche variabili che contengono stringhe letterali nel comando costruito alla funzione `quote_literal`. Entrambe le funzioni eseguono i passaggi appropriati per restituire il testo di input racchiuso tra virgolette doppie o singole rispettivamente, dove viene eseguito l'escape di qualsiasi carattere speciale incorporato.

Il Dollar quoting è utile solo per l'utilizzo di virgolette in testo fisso. Non scrivere l'esempio precedente nel seguente formato.

```
EXECUTE 'UPDATE tbl SET '
  || quote_ident(colname)
  || ' = $$'
  || newvalue
  || '$$ WHERE key = '
  || quote_literal(keyvalue);
```

Non esegui questa operazione perché l'esempio si interrompe se i contenuti di `newvalue` contengono $$. Lo stesso problema si applica a qualsiasi delimitatore di Dollar quoting che scegli. Per utilizzare in maniera sicura le virgolette in testo che non conosci in anticipo, utilizza la funzione `quote_literal`.

Return

L'istruzione RETURN ritorna all'intermediario da una procedura archiviata.

```
RETURN;
```

Di seguito viene riportato un esempio.

```
CREATE OR REPLACE PROCEDURE return_example(a int)
AS $$
BEGIN
  FOR b in 1..10 LOOP
    IF b < a THEN
      RAISE INFO 'b = %', b;
    ELSE
      RETURN;
    END IF;
  END LOOP;
END;
$$ LANGUAGE plpgsql;
```

Condizionali: IF

L'istruzione condizionale IF può avere le seguenti forme nel linguaggio PL/pgSQL utilizzato da Amazon Redshift:

- IF ... THEN

  ```
  IF boolean-expression THEN
    statements
  END IF;
  ```

  Di seguito viene riportato un esempio.

  ```
  IF v_user_id <> 0 THEN
    UPDATE users SET email = v_email WHERE user_id = v_user_id;
  END IF;
  ```

- IF ... THEN ... ELSE

  ```
  IF boolean-expression THEN
    statements
  ELSE
    statements
  END IF;
  ```

  Di seguito viene riportato un esempio.

  ```
  IF parentid IS NULL OR parentid = ''
  ```

```
THEN
  return_name = fullname;
  RETURN;
ELSE
  return_name = hp_true_filename(parentid) || '/' || fullname;
  RETURN;
END IF;
```

- IF ... THEN ... ELSIF ... THEN ... ELSE

  La parola chiave ELSIF si può anche scrivere come ELSEIF.

```
IF boolean-expression THEN
  statements
[ ELSIF boolean-expression THEN
  statements
[ ELSIF boolean-expression THEN
  statements
    ...] ]
[ ELSE
  statements ]
END IF;
```

  Di seguito viene riportato un esempio.

```
IF number = 0 THEN
  result := 'zero';
ELSIF number > 0 THEN
  result := 'positive';
ELSIF number < 0 THEN
  result := 'negative';
ELSE
  -- the only other possibility is that number is null
  result := 'NULL';
END IF;
```

Condizionali: CASE

L'istruzione condizionale CASE può avere le seguenti forme nel linguaggio PL/pgSQL utilizzato da Amazon Redshift:

- CASE semplice

```
CASE search-expression
WHEN expression [, expression [ ... ]] THEN
  statements
[ WHEN expression [, expression [ ... ]] THEN
  statements
  ... ]
[ ELSE
  statements ]
END CASE;
```

Una semplice istruzione CASE fornisce l'esecuzione condizionale basata sull'eguaglianza degli operandi.

Il valore *espressione di ricerca* viene valutato una volta e viene paragonato successivamente a ogni *espressione* nelle clausole WHEN. Se si trova una corrispondenza, vengono eseguite gli *statement* corrispondenti e il controllo passa all'istruzione successiva dopo END CASE. Le espressioni WHEN successive non vengono valutate. Se non viene trovata alcuna corrispondenza, vengono eseguite le *istruzioni* ELSE. Tuttavia, se ELSE non è presente, viene generata un'eccezione CASE_NOT_FOUND.

Di seguito viene riportato un esempio.

```
CASE x
WHEN 1, 2 THEN
  msg := 'one or two';
ELSE
  msg := 'other value than one or two';
END CASE;
```

- CASE ricercata

```
CASE
WHEN boolean-expression THEN
  statements
[ WHEN boolean-expression THEN
  statements
  ... ]
[ ELSE
  statements ]
```

```
END CASE;
```

La forma ricercata di CASE fornisce un'esecuzione condizionale basata sulla verità delle espressioni booleane.

Ogni *espressione booleana* della clausola WHEN viene valutata a turno, fino a che viene trovata una che genera true. In seguito le istruzioni corrispondenti vengono eseguite e il controllo passa all'istruzione successiva dopo END CASE. Le *espressioni* WHEN successive non vengono valutate. Se non viene trovato un risultato true , vengono eseguite le *istruzioni* ELSE. Tuttavia, se ELSE non è presente, viene generata un'eccezione CASE_NOT_FOUND.

Di seguito viene riportato un esempio.

```
CASE
WHEN x BETWEEN 0 AND 10 THEN
  msg := 'value is between zero and ten';
WHEN x BETWEEN 11 AND 20 THEN
  msg := 'value is between eleven and twenty';
END CASE;
```

Loop

Le istruzioni loop possono avere le seguenti forme nel linguaggio PL/pgSQL utilizzato da Amazon Redshift:

- Loop semplice

```
[<<label>>]
LOOP
  statements
END LOOP [ label ];
```

Un loop semplice definisce un loop non condizionale che viene ripetuto a tempo indeterminato fino a quando viene terminato da un'istruzione EXIT e RETURN. L'etichetta opzionale può essere utilizzata dalle istruzioni EXIT e CONTINUE nei loop nidificati per specificare a quale loop si riferiscono le instruzioni EXIT e CONTINUE.

Di seguito viene riportato un esempio.

```
CREATE OR REPLACE PROCEDURE simple_loop()
LANGUAGE plpgsql
AS $$
BEGIN
  <<simple_while>>
  LOOP
    RAISE INFO 'I am raised once';
    EXIT simple_while;
    RAISE INFO 'I am not raised';
  END LOOP;
  RAISE INFO 'I am raised once as well';
END;
$$;
```

- Loop exit

```
EXIT [ label ] [ WHEN expression ];
```

Se l'*etichetta* non è presente, il loop interno viene terminato e viene eseguita l'istruzione successiva a END LOOP. Se l'*etichetta* è presente, deve essere l'etichetta del livello corrente o del livello esterno del loop o blocco nidificato. In seguito, il loop o blocco denominato viene terminato e il controllo continua con l'istruzione dopo il loop o blocco corrispondente a END.

Se WHEN è specificata, l'uscita del loop avviene solo se l'*espressione* è true. Altrimenti, il controllo passa all'istruzione dopo EXIT.

È possibile utilizzare EXIT con tutti i tipi di loop, non è limitato all'utilizzo con loop non condizionali.

Quando viene utilizzato un blocco BEGIN, EXIT passa il controllo alla prossima istruzione dopo la fine del blocco. A questo scopo deve essere utilizzata un'etichetta. Un'istruzione EXIT non etichettata non viene mai considerata per un blocco BEGIN.

Di seguito viene riportato un esempio.

```
CREATE OR REPLACE PROCEDURE simple_loop_when(x int)
LANGUAGE plpgsql
AS $$
DECLARE i INTEGER := 0;
BEGIN
  <<simple_loop_when>>
```

```
  LOOP
    RAISE INFO 'i %', i;
    i := i + 1;
    EXIT simple_loop_when WHEN (i >= x);
  END LOOP;
END;
$$;
```

- Loop Continue

```
CONTINUE [ label ] [ WHEN expression ];
```

Se *label* non viene fornita, l'esecuzione passa alla successiva iterazione del loop più interno. Ovvero, tutte le istruzioni rimanenti nel testo del loop vengono ignorate. Il controllo in seguito ritorna all'espressione di controllo del loop (se presenti) per determinare se un'altra iterazione di un loop è necessaria. Se *label* è presente, specifica l'etichetta del loop la cui esecuzione viene continuata.

Se l'istruzione WHEN è specificata, l'iterazione successiva del loop inizia solo se *expression* è true. Altrimenti, il controllo passa all'istruzione dopo CONTINUE.

È possibile utilizzare CONTINUE con tutti i tipi di loop, non è limitato all'utilizzo con loop non condizionali.

```
CONTINUE mylabel;
```

- Loop WHILE

```
[<<label>>]
WHILE expression LOOP
  statements
END LOOP [ label ];
```

L'istruzione WHILE ripete una sequenza di istruzioni purché l'*boolean-expression* corrisponda a true. L'espressione viene controllata appena prima di ogni inserimento nel body del loop.

Di seguito viene riportato un esempio.

```
WHILE amount_owed > 0 AND gift_certificate_balance > 0 LOOP
  -- some computations here
END LOOP;
```

```
WHILE NOT done LOOP
  -- some computations here
END LOOP;
```

- Loop FOR (variante numero intero)

```
[<<label>>]
FOR name IN [ REVERSE ] expression .. expression LOOP
  statements
END LOOP [ label ];
```

Il loop FOR (variante numero intero) crea un loop che esegue l'iterazione in una gamma di valori interi. Il nome variabile viene definito automaticamente come tipo intero ed esiste solo nel loop. Qualsiasi definizione esistente del nome della variabile viene ignorato nel loop. Le due espressioni che forniscono il limite inferiore e superiore della gamma vengono valutate una volta quando entrano nel loop. Se si specifica REVERSE, viene sottratto il valore di incremento, piuttosto che aggiunto, dopo ogni iterazione.

Se il limite inferiore è maggiore rispetto al limite superiore (o inferiore, nel caso REVERSE), il body del loop non viene eseguito. Non viene generato alcun errore.

Se un'etichetta viene collegata al loop FOR, puoi fare riferimento alla variabile del loop intero con un nome qualificato, utilizzando quell'etichetta.

Di seguito viene riportato un esempio.

```
FOR i IN 1..10 LOOP
  -- i will take on the values 1,2,3,4,5,6,7,8,9,10 within the loop
END LOOP;

FOR i IN REVERSE 10..1 LOOP
  -- i will take on the values 10,9,8,7,6,5,4,3,2,1 within the loop
END LOOP;
```

- Loop FOR (variante set di risultati)

```
[<<label>>]
FOR target IN query LOOP
  statements
END LOOP [ label ];
```

Il *target* è una variabile di record o un elenco separato da virgola di variabili scalari. In seguito, al target viene assegnata ciascuna riga risultante dalla query e il body del loop viene eseguito per ogni riga.

Il loop FOR (variante set di risultati) permette a una procedura archiviata di effettuare l'iterazione attraverso i risultati di una query e manipolare i dati di conseguenza.

Di seguito viene riportato un esempio.

```
CREATE PROCEDURE cs_refresh_reports() AS $$
DECLARE
  reports RECORD;
BEGIN
  FOR reports IN SELECT * FROM cs_reports ORDER BY sort_key LOOP
    -- Now "reports" has one record from cs_reports
    EXECUTE 'INSERT INTO ' || quote_ident(reports.report_name) || ' ' ||
 reports.report_query;
  END LOOP;
  RETURN;
END;
$$ LANGUAGE plpgsql;
```

- Loop FOR con SQL dinamico

```
[<<label>>]
FOR record_or_row IN EXECUTE text_expression LOOP
  statements
END LOOP;
```

Un loop FOR con SQL dinamico permette a una procedura archiviata di effettuare l'iterazione attraverso i risultati di una query dinamica e manipolare i dati di conseguenza.

Di seguito viene riportato un esempio.

```
CREATE OR REPLACE PROCEDURE for_loop_dynamic_sql(x int)
LANGUAGE plpgsql
AS $$
DECLARE
  rec RECORD;
  query text;
BEGIN
```

```
  query := 'SELECT * FROM tbl_dynamic_sql LIMIT ' || x;
  FOR rec IN EXECUTE query
  LOOP
    RAISE INFO 'a %', rec.a;
  END LOOP;
END;
$$;
```

Cursori

Invece di eseguire una query intera in una volta, puoi impostare un cursore. Un cursore incapsula una query e legge il risultato della query un po' di righe alla volta. Un motivo è di evitare il superamento della memoria quando il risultato contiene un ampio numero di righe. Un altro motivo è restituire un riferimento a un cursore che una procedura archiviata ha creato, che permette all'intermediario di leggere le righe. Questo approccio fornisce un modo efficace di restituire set di righe di grandi dimensioni dalle procedure archiviate.

Per utilizzare i cursori in una procedura archiviata NONATOMIC, posizionare il loop del cursore tra START TRANSACTION...COMMIT.

Per impostare un cursore, è necessario prima dichiarare un variabile di cursore. Tutto l'accesso ai cursori in PL/pgSQL passa tramite le variabili di cursore, che sono sempre del tipo di dati speciali `refcursor`. Un tipo di dati `refcursor` semplicemente contiene un riferimento a un cursore.

Puoi creare una variabile di cursore dichiarandola come tipo di variabile `refcursor`. Oppure puoi utilizzare la seguente sintassi di dichiarazione del cursore.

```
name CURSOR [ ( arguments ) ] FOR query ;
```

Nell'esempio precedente, *`arguments`* (se specificato) è un elenco separato da virgole delle coppie *`name datatype`* dove ognuna definisce i nome da sostituite con i valori di parametro nella *`query`*. I valori effettivi che sostituiscono questi nomi vengono specificati più avanti, quando il cursore è aperto.

Di seguito vengono riportati degli esempi.

```
DECLARE
  curs1 refcursor;
  curs2 CURSOR FOR SELECT * FROM tenk1;
  curs3 CURSOR (key integer) IS SELECT * FROM tenk1 WHERE unique1 = key;
```

Tutte e tre queste variabili hanno il tipo di dati `refcursor`, ma la prima può essere utilizzata con qualsiasi query. Al contrario, il secondo ha una query pienamente specificata già associata e l'ultima ha una query parametrizzata associata. Il valore `key` è sostituito da un valore di parametro intero quando il cursore è aperto. La variabile `curs1` viene definita non vincolata perché non è vincolata ad alcuna query particolare.

Prima di poter utilizzare un cursore per recuperare righe, deve essere aperto. PL/pgSQL ha tre forma di istruzione OPEN, delle quali due utilizzano variabili di cursore non vincolate e la terza utilizza una variabile di cursore vincolata:

- Open for select: la variabile del cursore viene aperta e riceve la query specificata da eseguire. Il cursore non può essere già aperto. Inoltre, deve essere stata dichiarata come cursore non vincolato (ovvero come variabile semplice `refcursor`). La query SELECT viene trattata allo stesso modo delle altre istruzioni SELECT in PL/pgSQL.

  ```
  OPEN cursor_name FOR SELECT ...;
  ```

  Di seguito viene riportato un esempio.

  ```
  OPEN curs1 FOR SELECT * FROM foo WHERE key = mykey;
  ```

- Open for execute: la variabile del cursore viene aperta e riceve la query specificata da eseguire. Il cursore non può essere già aperto. Inoltre, deve essere stata dichiarata come cursore non vincolato (ovvero come variabile semplice `refcursor`). La query è specificata come una espressione di stringa allo stesso modo del comando EXECUTE. Questo approccio fornisce flessibilità in modo che la query possa variare da un'esecuzione all'altra.

  ```
  OPEN cursor_name FOR EXECUTE query_string;
  ```

  Di seguito viene riportato un esempio.

  ```
  OPEN curs1 FOR EXECUTE 'SELECT * FROM ' || quote_ident($1);
  ```

- Open a bound cursor: questa forma di OPEN viene utilizzata per aprire una variabile di cursore la quale query vi era associata al momento della dichiarazione. Il cursore non può essere già aperto. Un elenco di espressioni di valore dell'argomento deve apparire se e solo se il cursore può accettare argomenti. Questi valori vengono sostituiti nella query.

```
OPEN bound_cursor_name [ ( argument_values ) ];
```

Di seguito viene riportato un esempio.

```
OPEN curs2;
OPEN curs3(42);
```

Dopo l'apertura di un cursore, puoi lavorarci utilizzando le istruzioni descritte di seguito. Queste istruzioni non devono aver luogo nella stessa procedura archiviata che ha aperto il cursore. Puoi restituire un valore `refcursor` da una procedura archiviata e lasciare che l'intermediario lavori sul cursore. Tutti i portali sono chiusi implicitamente alla fine della transazione. In seguito, puoi utilizzare un valore `refcursor` per far riferimento a un cursore aperto solo fino alla fine della transazione.

- FETCH recupera la riga successiva dal cursore in un target. Questo target può essere una variabile di riga, una variabile di record o un elenco separato da virgole di variabili semplici, come in SELECT INTO. Come con SELECT INTO; puoi controllare la variabile speciale FOUND per vedere se è stata ottenuta una riga.

  ```
  FETCH cursor INTO target;
  ```

  Di seguito viene riportato un esempio.

  ```
  FETCH curs1 INTO rowvar;
  ```

- CLOSE chiude il portale sotto un cursore aperto. È possibile utilizzare questa istruzione per rilasciare le risorse prima della fine della transazione. Puoi anche utilizzare questa istruzione per permettere alla variabile del cursore di essere nuovamente aperta.

  ```
  CLOSE cursor;
  ```

  Di seguito viene riportato un esempio.

  ```
  CLOSE curs1;
  ```

RAISE

Utilizza l'istruzione `RAISE level` per segnalare messaggi e generare errori.

```
RAISE level 'format' [, variable [, ...]];
```

I livelli possibili sono NOTICE, INFO, LOG, WARNING e EXCEPTION. EXCEPTION genera un errore, che normalmente annulla la transazione corrente. Gli altri livelli generano solo messaggi di diversi livelli di priorità.

Nella stringa formato, % viene sostituito dalla rappresentazione di stringa dell'argomento successivo opzionale. Scrivi %% per emetter un % letterale. Al momento, gli argomenti opzionali devono essere semplici variabili, non espressioni, e il formato deve essere un valore letterale di stringa semplice.

Nell'esempio seguente, il valore di `v_job_id` sostituisce % nella stringa.

```
RAISE NOTICE 'Calling cs_create_job(%)', v_job_id;
```

Utilizza l'istruzione `RAISE` per generare nuovamente l'eccezione rilevata da un blocco di gestione delle eccezioni. Questa istruzione è valida solo nei blocchi di gestione delle eccezioni delle procedure archiviate in modalità NONATOMIC.

```
RAISE;
```

Controllo della transazione

Utilizzare le istruzioni di controllo delle transazioni nel linguaggio PL/pgSQL utilizzato da Amazon Redshift. Per informazioni sull'utilizzo delle dichiarazioni COMMIT, ROLLBACK e TRUNCATE, all'interno di una procedura archiviata, consultare Gestione delle transazioni.

Nelle procedure archiviate in modalità NONATOMIC, utilizza `START TRANSACTION` per avviare un blocco di transazioni.

```
START TRANSACTION;
```

> **Note**
>
> Di seguito sono indicate le differenze tra l'istruzione PL/pgSQL START TRANSACTION e il comando SQL START TRANSACTION:

- All'interno delle procedure archiviate, START TRANSACTION non è sinonimo di BEGIN.
- L'istruzione PL/pgSQL non supporta il livello di isolamento opzionale e le parole chiave di autorizzazione di accesso.

Creazione di viste materializzate in Amazon Redshift

In un ambiente di data warehouse, le applicazioni spesso devono eseguire query complesse su tabelle di grandi dimensioni. Un esempio sono le istruzioni SELECT che eseguono unioni e aggregazioni multitabella su tabelle che contengono miliardi di righe. L'elaborazione di queste query può essere costosa in termini di risorse di sistema e di tempo necessario per ottenere i risultati.

Le viste materializzate in Amazon Redshift offrono un modo per affrontare questi problemi. Una vista materializzata contiene un set di risultati pre-elaborato, basato su una query SQL su una o più tabelle di base. È possibile eseguire istruzioni SELECT per eseguire una query su una vista materializzata allo stesso modo in cui è possibile eseguire query su altre tabelle o viste del database. Amazon Redshift restituisce i risultati pre-elaborati dalla vista materializzata senza dover accedere in alcun modo alle tabelle di base. Dal punto di vista dell'utente, i risultati della query vengono restituiti molto più rapidamente rispetto al recupero degli stessi dati dalle tabelle di base.

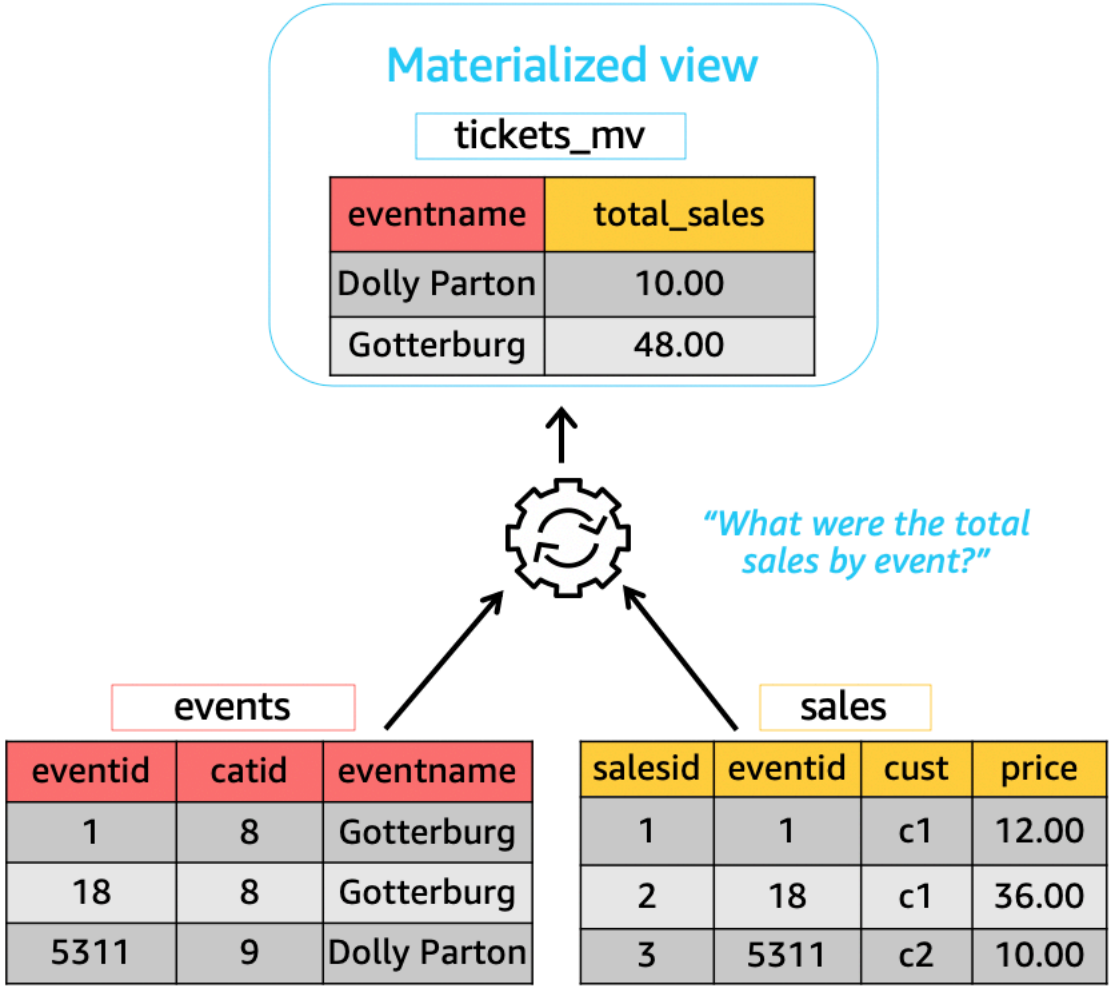
Le viste materializzate sono particolarmente utili per velocizzare le query prevedibili e ripetute. Invece di eseguire query ad alto uso di risorse su tabelle di database di grandi dimensioni (come aggregazioni o più join), le applicazioni possono eseguire query su una vista materializzata e recuperare un set di risultati pre-elaborato. Ad esempio, considera lo scenario in cui viene utilizzata una serie di query per popolare le dashboard, come Amazon. QuickSight Questo caso d'uso è ideale per una vista materializzata, perché le query sono prevedibili e ripetute più e più volte.

È possibile definire una vista materializzata in termini di altre viste materializzate. Utilizzare viste materializzate su viste materializzate per espandere la capacità delle viste materializzate. In questo approccio, una vista materializzata esistente svolge lo stesso ruolo di una tabella di base per il recupero dei dati da parte della query.

Questo approccio è particolarmente utile per riutilizzare i join precalcolati per diverse opzioni aggregate o GROUP BY. Ad esempio, è possibile utilizzare una vista materializzata che unisce le informazioni del cliente (contenenti milioni di righe) alle informazioni dettagliate sull'ordine di un articolo (contenenti miliardi di righe). Si tratta di una query costosa da calcolare ripetutamente on demand. È possibile utilizzare diverse opzioni GROUP BY per le viste materializzate create sopra questa vista materializzata e unirsi ad altre tabelle. In questo modo si risparmia tempo di calcolo altrimenti utilizzato per eseguire il costoso join sottostante ogni volta. La tabella STV_MV_DEPS riporta le dipendenze di una vista materializzata su altre viste materializzate.

Quando si crea una vista materializzata, Amazon Redshift esegue l'istruzione SQL specificata dall'utente per raccogliere i dati dalla tabella o dalle tabelle di base e memorizzare il set di risultati.

Nella figura seguente viene fornita una panoramica della vista `tickets_mv` materializzata definita da una query SQL utilizzando due tabelle di base `events` e `sales`.



È quindi possibile utilizzare queste viste materializzate nelle query per velocizzarle. Inoltre, Amazon Redshift può riscrivere automaticamente queste query per utilizzare le viste materializzate, anche quando la query non fa esplicito riferimento a una vista materializzata. La riscrittura automatica delle query è particolarmente efficace per migliorare le prestazioni quando non è possibile modificare le query per utilizzare le viste materializzate.

Per aggiornare i dati nella vista materializzata, è possibile utilizzare l'istruzione REFRESH MATERIALIZED VIEW in qualsiasi momento per aggiornare manualmente le viste materializzate. Amazon Redshift identifica le modifiche apportate alla tabella o alle tabelle di base e quindi applica tali modifiche alla vista materializzata. Poiché la riscrittura automatica delle query richiede che le viste materializzate siano aggiornate , in qualità di proprietario delle viste materializzate, assicurarsi di aggiornare le viste ogni volta che viene modificata una tabella di base.

Amazon Redshift fornisce alcuni metodi per mantenere aggiornate le viste materializzate per la riscrittura automatica. È possibile configurare le viste materializzate con l'opzione di aggiornamento automatico per aggiornare le viste materializzate quando le tabelle di base delle viste materializzate vengono aggiornate. Questa operazione di aggiornamento automatico viene eseguita in un momento in cui le risorse del cluster sono disponibili per ridurre al minimo le interruzioni di altri carichi di lavoro. Poiché la pianificazione dell'aggiornamento automatico dipende dal carico di lavoro, è possibile avere un maggiore controllo sul momento in cui Amazon Redshift aggiorna le viste materializzate. È possibile pianificare un processo di aggiornamento delle viste materializzate utilizzando l'API del pianificatore di Amazon Redshift e l'integrazione della console. Per ulteriori informazioni sulla pianificazione delle query, consultare Pianificazione di una query sulla console Amazon Redshift.

Questa operazione è particolarmente utile quando esiste un requisito di accordo sul livello di servizio (SLA) per up-to-date i dati da una vista materializzata. È inoltre possibile aggiornare manualmente tutte le viste materializzate che è possibile aggiornare automaticamente. Per informazioni su come creare viste materializzate, consultare CREATE MATERIALIZED VIEW.

Puoi emettere istruzioni SELECT per eseguire query su una vista materializzata. Per informazioni su come eseguire query sulle viste materializzate, consultare Query di una vista materializzata.. Dopo un certo periodo di tempo il set di risultati diventa obsoleto, quando altri dati vengono inseriti, aggiornati ed eliminati nelle tabelle di base. Puoi aggiornare la vista materializzata in qualsiasi momento per aggiornarla con le ultime modifiche delle tabelle di base. Per informazioni su come aggiornare le viste materializzate, consultare REFRESH MATERIALIZED VIEW.

Per informazioni sui comandi SQL per creare e gestire le viste materializzate, consultare i seguenti argomenti sui comandi:

- CREATE MATERIALIZED VIEW
- ALTER MATERIALIZED VIEW
- REFRESH MATERIALIZED VIEW
- DROP MATERIALIZED VIEW

Per informazioni sulle tabelle di sistema e le viste per monitorare le viste materializzate, consultare i seguenti argomenti:

- STV_MV_INFO
- STL_MV_STATE
- SVL_MV_REFRESH_STATUS

- STV_MV_DEPS

Argomenti

- Query di una vista materializzata.
- Riscrittura automatica delle query per utilizzare le viste materializzate
- Aggiornamento di una vista materializzata
- Viste materializzate automatizzate
- Utilizzo di una funzione definita dall'utente (UDF) in una vista materializzata
- Importazione di dati in streaming

Query di una vista materializzata.

È possibile utilizzare una vista materializzata in qualsiasi query SQL facendo riferimento al nome della vista materializzata come origine dati, come nel caso di una tabella o di una vista standard.

Quando una query accede a una vista materializzata, vede esclusivamente i dati memorizzati nella vista materializzata relativi all'ultimo aggiornamento. Perciò la query potrebbe non vedere tutte le ultime modifiche apportate alle corrispondenti tabelle di base della vista materializzata.

Se altri utenti desiderano eseguire delle query sulla vista materializzata, il proprietario della vista materializzata deve concedere l'autorizzazione SELECT a tali utenti. Gli altri utenti non hanno bisogno dell'autorizzazione SELECT sulle tabelle di base sottostanti. Il proprietario della vista materializzata può anche revocare l'autorizzazione SELECT agli altri utenti per impedire che possano eseguire query sulla vista materializzata.

Se il proprietario della vista materializzata non dispone più dell'autorizzazione SELECT sulle tabelle di base sottostanti:

- Il proprietario non può più eseguire query sulla vista materializzata.
- Gli altri utenti che dispongono dell'autorizzazione SELECT sulla vista materializzata non possono più eseguire query su di essa.

Nell'esempio seguente viene eseguita una query sulla vista materializzata `tickets_mv` Per ulteriori informazioni sul comando SQL utilizzato per creare una vista materializzata, consultare CREATE MATERIALIZED VIEW.

```
SELECT sold
FROM tickets_mv
WHERE catgroup = 'Concerts';
```

Poiché i risultati della query sono pre-calcolati, non è necessario accedere alle tabelle sottostanti (`category`, `event` e `sales`). Amazon Redshift può restituire i risultati direttamente da `tickets_mv`.

Riscrittura automatica delle query per utilizzare le viste materializzate

È possibile utilizzare la riscrittura automatica delle query delle viste materializzate in Amazon Redshift per consentire ad Amazon Redshift di riscrivere le query per utilizzare le viste materializzate. Questa operazione consente di accelerare i carichi di lavoro delle query anche per le query che non fanno riferimento esplicitamente a una vista materializzata. Quando Amazon Redshift riscrive le query, utilizza solo viste materializzate aggiornate .

Note per l'utilizzo

Per verificare se per una query viene utilizzata la riscrittura automatica delle query, è possibile esaminare il piano di query o STL_EXPLAIN. Di seguito sono illustrati un'istruzione SELECT e l'output EXPLAIN del piano di query originale.

```
SELECT catgroup, SUM(qtysold) AS sold
FROM category c, event e, sales s
WHERE c.catid = e.catid AND e.eventid = s.eventid
GROUP BY 1;

EXPLAIN
 XN HashAggregate  (cost=920021.24..920021.24 rows=1 width=35)
   ->  XN Hash Join DS_BCAST_INNER  (cost=440004.53..920021.22 rows=4 width=35)
         Hash Cond: ("outer".eventid = "inner".eventid)
         ->  XN Seq Scan on sales s  (cost=0.00..7.40 rows=740 width=6)
         ->  XN Hash  (cost=440004.52..440004.52 rows=1 width=37)
               ->  XN Hash Join DS_BCAST_INNER  (cost=0.01..440004.52 rows=1 width=37)
                     Hash Cond: ("outer".catid = "inner".catid)
                     ->  XN Seq Scan on event e  (cost=0.00..2.00 rows=200 width=6)
                     ->  XN Hash  (cost=0.01..0.01 rows=1 width=35)
```

```
                          ->  XN Seq Scan on category c  (cost=0.00..0.01 rows=1
 width=35)
```

Quanto segue mostra l'output EXPLAIN dopo una riscrittura automatica riuscita. Questo output include una scansione della vista materializzata nel piano di query che sostituisce parti del piano di query originale.

```
* EXPLAIN
     XN HashAggregate  (cost=11.85..12.35 rows=200 width=41)
       ->  XN Seq Scan on mv_tbl__tickets_mv__0 derived_table1  (cost=0.00..7.90
 rows=790 width=41)
```

Per la riscrittura automatica delle query vengono prese in considerazione solo le up-to-date (nuove) viste materializzate, indipendentemente dalla strategia di aggiornamento, ad esempio automatica, pianificata o manuale. Pertanto, la query originale restituisce risultati. up-to-date Quando nelle query viene esplicitamente fatto riferimento a una vista materializzata, Amazon Redshift accede ai dati attualmente memorizzati nella vista materializzata. Questi dati potrebbero non riflettere le ultime modifiche delle tabelle di base della vista materializzata.

È possibile utilizzare la riscrittura automatica delle query delle viste materializzate create nella versione del cluster 1.0.20949 o successiva.

È possibile arrestare la riscrittura automatica delle query a livello di sessione utilizzando SET mv_enable_aqmv_for_session impostato su FALSE.

Limitazioni

Di seguito sono riportate le limitazioni per l'utilizzo della riscrittura automatica delle query delle viste materializzate:

- La riscrittura automatica delle query funziona con viste materializzate che non fanno riferimento o includono uno dei seguenti elementi:
  - Sottoquery
  - Join left, right o full outer
  - Operazioni del set
  - Qualsiasi funzione di aggregazione, tranne SUM, COUNT, MIN, MAX e AVG. (Queste sono le uniche funzioni aggregate che funzionano con la riscrittura automatica delle query.)
  - Qualsiasi funzione aggregata con DISTINCT

  - Qualsiasi funzione finestra
  - Clausole SELECT DISTINCT o HAVING
  - Tabelle esterna
  - Altre viste materializzate
- La riscrittura automatica delle query riscrive le query SELECT che fanno riferimento alle tabelle Amazon Redshift definite dall'utente. Amazon Redshift non riscrive le seguenti query:
  - Istruzioni CREATE TABLE AS
  - Istruzioni SELECT INTO
  - Query su cataloghi o tabelle di sistema
  - Query con join esterni o una clausola SELECT DISTINCT
- Se una query non viene riscritta automaticamente, verifica se disponi dell'autorizzazione SELECT sulla vista materializzata specificata e che l'opzione mv_enable_aqmv_for_session sia impostata su TRUE.

  È inoltre possibile verificare se le viste materializzate sono idonee per la riscrittura automatica delle query ispezionando STV_MV_INFO. Per ulteriori informazioni, consultare STV_MV_INFO.

Aggiornamento di una vista materializzata

Quando crei una vista materializzata, il suo contenuto riflette lo stato della tabella o delle tabelle del database sottostante in quel momento. I dati nella vista materializzata rimangono invariati, anche quando le applicazioni apportano modifiche ai dati nelle tabelle sottostanti. Per aggiornare i dati nella vista materializzata, è possibile utilizzare l'istruzione `REFRESH MATERIALIZED VIEW` in qualsiasi momento per aggiornare manualmente le viste materializzate. Quando si utilizza questa istruzione, Amazon Redshift identifica le modifiche apportate nella tabella o nelle tabelle di base e quindi applica tali modifiche alla vista materializzata.

Amazon Redshift dispone di due strategie per aggiornare una vista materializzata:

- In molti casi, Amazon Redshift è in grado di eseguire un aggiornamento incrementale. In un aggiornamento incrementale, Amazon Redshift identifica rapidamente le modifiche ai dati nelle tabelle di base dall'ultimo aggiornamento e aggiorna i dati nella vista materializzata. L'aggiornamento incrementale è supportato o segue costrutti SQL utilizzati nella query durante la definizione della vista materializzata.

  - Costrutti che contengono le clausole SELECT, FROM, [INNER] JOIN, WHERE, GROUP BY, HAVING.
  - Costrutti contenenti aggregazioni, come SUM, MIN, MAX, AVG e COUNT.
  - Molte funzioni SQL integrate, in particolare quelle immutabili, forniscono gli stessi argomenti di input e producono sempre lo stesso output.

  L'aggiornamento incrementale è supportato anche per una vista materializzata basata su una tabella di unità di condivisione dati.
- Se non è possibile eseguire un aggiornamento incrementale, allora Amazon Redshift esegue un aggiornamento completo. Un aggiornamento completo esegue nuovamente il comando SQL sottostante, sostituendo tutti i dati della vista materializzata.
- Amazon Redshift sceglie automaticamente il metodo di aggiornamento per una vista materializzata in base alla query SELECT utilizzata per definire la vista materializzata.

L'aggiornamento di una vista materializzata su una vista materializzata non è un processo a cascata. In altre parole, supponiamo di avere una vista materializzata A che dipende dalla vista materializzata B. In questo caso, quando viene richiamato REFRESH MATERIALIZED VIEW A, A viene aggiornato utilizzando la versione corrente di B, anche quando B lo è. out-of-date Perché A sia completamente aggiornata , prima di aggiornare A, aggiornare B in una transazione separata.

L'esempio seguente mostra come creare un piano di aggiornamento completo per una vista materializzata a livello di programmazione. Per aggiornare la vista materializzata v, aggiornare prima la vista materializzata u. Per aggiornare la vista materializzata w, aggiornare prima la vista materializzata u e poi la vista materializzata v.

```
CREATE TABLE t(a INT);
CREATE MATERIALIZED VIEW u AS SELECT * FROM t;
CREATE MATERIALIZED VIEW v AS SELECT * FROM u;
CREATE MATERIALIZED VIEW w AS SELECT * FROM v;

WITH RECURSIVE recursive_deps (mv_tgt, lvl, mv_dep) AS
( SELECT trim(name) as mv_tgt, 0 as lvl, trim(ref_name) as mv_dep
  FROM stv_mv_deps
  UNION ALL
  SELECT R.mv_tgt, R.lvl+1 as lvl, trim(S.ref_name) as mv_dep
  FROM stv_mv_deps S, recursive_deps R
  WHERE R.mv_dep = S.name
)
```

```
SELECT mv_tgt, mv_dep from recursive_deps
ORDER BY mv_tgt, lvl DESC;

 mv_tgt | mv_dep
--------+--------
 v      | u
 w      | u
 w      | v
(3 rows)
```

L'esempio seguente mostra un messaggio informativo quando si esegue REFRESH MATERIALIZED VIEW su una vista materializzata che dipende da una vista materializzata. out-of-date

```
create table a(a int);
```

```
create materialized view b as select * from a;
```

```
create materialized view c as select * from b;
```

```
insert into a values (1);
```

```
refresh materialized view c;

INFO:  Materialized view c is already up to date.  However, it depends on another
 materialized view that is not up to date.
```

```
REFRESH MATERIALIZED VIEW b;
INFO:  Materialized view b was incrementally updated successfully.
```

```
REFRESH MATERIALIZED VIEW c;
INFO:  Materialized view c was incrementally updated successfully.
```

Amazon Redshift presenta attualmente le seguenti limitazioni per l'aggiornamento incrementale delle viste materializzate.

Amazon Redshift attualmente non supporta l'aggiornamento incrementale per le viste materializzate definite con una query utilizzando uno dei seguenti elementi SQL:

- OUTER JOIN (RIGHT, LEFT o FULL).
- Operazioni di set: UNION, INTERSECT, EXCEPT e MINUS
- Le funzioni di aggregazione MEDIAN, PERCENTILE_CONT, LISTAGG, STDDEV_SAMP, STDDEV_POP, APPROXIMATE COUNT, APPROXIMATE PERCENTILE e le funzioni di aggregazione bitwise

  > **Note**
  >
  > Sono supportate le funzioni di aggregazione COUNT, SUM e AVG.

- Funzioni di aggregazione di tipo DISTINCT, come DISTINCT COUNT, DISTINCT SUM, ecc.
- Funzioni finestra
- Query che utilizza tabelle temporanee per l'ottimizzazione delle query, ad esempio l'ottimizzazione delle espressioni secondarie comuni.
- Sottointerrogazioni.
- Tabelle esterne che fanno riferimento ai seguenti formati nella query che definisce la vista materializzata.
  - Delta Lake
  - Hudi

  L'aggiornamento incrementale è supportato nella traccia di anteprima per le viste materializzate definite utilizzando formati diversi dai precedenti. Per ulteriori informazioni sulla configurazione dei cluster di anteprima, consulta Creazione di un cluster di anteprima nella Guida alla gestione di Amazon Redshift. Per informazioni sulla configurazione dei gruppi di lavoro di anteprima, consulta Creazione di un gruppo di lavoro di anteprima nella Guida alla gestione di Amazon Redshift.

Aggiornamento automatico di una vista materializzata

Amazon Redshift può aggiornare automaticamente le viste materializzate con up-to-date i dati delle sue tabelle di base quando le viste materializzate vengono create o modificate per avere l'opzione di aggiornamento automatico. Dopo le modifiche alle tabelle di base, Amazon Redshift aggiorna automaticamente le viste materializzate il prima possibile.

Per completare l'aggiornamento delle viste materializzate più importanti con un impatto minimo sui carichi di lavoro attivi nel cluster, Amazon Redshift considera diversi fattori. Questi fattori includono il

carico di sistema corrente, le risorse necessarie per l'aggiornamento, le risorse cluster disponibili e la frequenza di utilizzo delle viste materializzate.

Amazon Redshift assegna la priorità ai carichi di lavoro rispetto all'aggiornamento automatico e potrebbe interrompere l'aggiornamento automatico per preservare le prestazioni del carico di lavoro degli utenti. Questo approccio potrebbe ritardare l'aggiornamento di alcune viste materializzate. In alcuni casi, è possibile che sia necessario usare un comportamento di aggiornamento più deterministico per le viste materializzate. In tal caso, valutare l'utilizzo dell'aggiornamento manuale come descritto in REFRESH MATERIALIZED VIEW o l'aggiornamento pianificato utilizzando le operazioni API del pianificatore Amazon Redshift o la console.

È possibile impostare l'aggiornamento automatico per le viste materializzate utilizzando CREATE MATERIALIZED VIEW. È inoltre possibile utilizzare la clausola AUTO REFRESH per aggiornare automaticamente le viste materializzate. Per ulteriori informazioni sulla creazione di viste materializzate, consultare CREATE MATERIALIZED VIEW. È possibile abilitare l'aggiornamento automatico per una vista materializzata corrente utilizzando ALTER MATERIALIZED VIEW.

Durante l'aggiornamento delle viste materializzate, considerare quanto riportato di seguito:

- È comunque possibile aggiornare esplicitamente una vista materializzata utilizzando il comando REFRESH MATERIALIZED VIEW anche se non è stato abilitato l'aggiornamento automatico per la vista materializzata.
- Amazon Redshift non aggiorna automaticamente le viste materializzate definite nelle tabelle esterne.
- Per lo stato di aggiornamento, è possibile controllare SVL_MV_REFRESH_STATUS, che registra le query avviate dall'utente o aggiornate automaticamente.
- Per eseguire REFRESH sulle viste materializzate di solo ricalcolo, assicurarsi di disporre dell'autorizzazione CREATE per gli schemi. Per ulteriori informazioni, consulta GRANT.

Viste materializzate automatizzate

Le viste materializzate sono un potente strumento per migliorare le prestazioni delle query in Amazon Redshift. Lo fanno memorizzando un set di risultati precalcolato. Query simili non devono eseguire nuovamente la stessa logica ogni volta, perché possono recuperare record dal set di risultati esistente. Gli sviluppatori e gli analisti creano le viste materializzate dopo aver analizzato i loro carichi di lavoro per determinare quali query trarrebbero beneficio e se il costo di manutenzione di ogni vista materializzata vale la pena. Man mano che i carichi di lavoro aumentano o cambiano, queste viste

materializzate devono essere revisionate per garantire che continuino a fornire vantaggi concreti in termini di prestazioni.

La funzionalità AutoV (Automated Materialized Views) di Redshift offre gli stessi vantaggi prestazionali delle viste materializzate create dall'utente. Amazon Redshift monitora continuamente il carico di lavoro utilizzando il machine learning e quindi crea nuove viste materializzate quando sono utili. AutoMV bilancia i costi di creazione e mantenimento aggiornati delle viste materializzate rispetto ai vantaggi previsti per la latenza delle query. Il sistema controlla anche gli AutoMV creati in precedenza e li elimina quando non sono più utili.

Il comportamento e le funzionalità di AutoMV sono gli stessi delle viste materializzate create dall'utente. Vengono aggiornati automaticamente e incrementalmente, utilizzando gli stessi criteri e restrizioni. Proprio come le viste materializzate create dagli utenti, Riscrittura automatica delle query per utilizzare le viste materializzate identifica le query che possono trarre vantaggio dagli AutoMV creati dal sistema. e le riscrive automaticamente per utilizzare gli AutoMV, migliorando le prestazioni delle query. Gli sviluppatori non devono modificare le query per sfruttare AutoMV.

> **Note**
>
> Le viste materializzate automatizzate vengono aggiornate in modo intermittente. Le query riscritte per utilizzare AutoMV restituiscono sempre i risultati più recenti. Quando Redshift rileva che i dati non sono aggiornati, le query non vengono riscritte per eseguire la lettura da viste materializzate automatizzate. Le query selezionano invece i dati più recenti dalle tabelle di base.

Qualsiasi carico di lavoro con query utilizzate ripetutamente può trarre vantaggio da AutoMV. Casi di utilizzo comune comprendono:

- Pannelli di controllo- I pannelli di controllo sono ampiamente utilizzati per fornire visualizzazioni rapide di indicatori aziendali chiave (KPI), eventi, tendenze e altri parametri. Spesso hanno un layout comune con grafici e tabelle, ma mostrano viste diverse per il filtraggio o per le operazioni di selezione delle dimensioni, come il drilldown. I pannelli di controllo hanno spesso un insieme comune di query utilizzate ripetutamente con parametri diversi. Le query del pannello di controllo possono trarre grande vantaggio dalle viste materializzate automatizzate.
- Report- Le query di segnalazione possono essere programmate a diverse frequenze, in base ai requisiti aziendali e al tipo di segnalazione. Inoltre, possono essere automatizzate o on demand.

Una caratteristica comune delle query di segnalazione è che possono avere una lunga durata e richiedono un uso intensivo di risorse. Con AutoMV, queste query non devono essere ricalcolate ogni volta che vengono eseguite, riducendo il tempo di esecuzione per ogni query e l'utilizzo delle risorse in Redshift.

Per disattivare le viste materializzate automatizzate, aggiornare il gruppo di parametri `auto_mv` a `false`. Per ulteriori informazioni, consultare Gruppi di parametri di Amazon Redshift nella Guida alla gestione dei cluster Amazon Redshift.

Ambito SQL e considerazioni per le viste materializzate automatizzate

- È possibile attivare e creare una vista materializzata da una query o da una subquery, purché contenga una clausola `GROUP BY` o una delle seguenti funzioni di aggregazione: SUM, COUNT, MIN, MAX o AVG. Ma non può contenere quanto segue:
  - Join left, right o full outer
  - Funzioni di aggregazione diverse da SUM, COUNT, MIN, MAX e AVG. (Queste particolari funzioni operano con la riscrittura automatica delle query.)
  - Qualsiasi funzione di aggregazione che includa DISTINCT
  - Qualsiasi funzione finestra
  - Clausole SELECT DISTINCT o HAVING
  - Altre viste materializzate

  Non è garantito che una query che soddisfa i criteri attivi la creazione di una vista materializzata automatizzata. Il sistema determina da quali candidati creare una vista, in base al vantaggio previsto per il carico di lavoro e al costo in termini di risorse da gestire, che include il costo per l'aggiornamento del sistema. Ogni vista materializzata risultante può essere utilizzata dalla riscrittura automatica delle query.
- Anche se AutoMV potrebbe essere attivato da una subquery o da singoli operatori di definizione, la vista materializzata risultante non conterrà subquery o operatori di definizione.
- Per determinare se AutoMV è stato utilizzato per le query, visualizzare il piano EXPLAIN e cercare `%_auto_mv_%` nell'output. Per ulteriori informazioni, consulta EXPLAIN.
- Le viste materializzate automatizzate non sono supportate nelle tabelle esterne, come unità di condivisione dati e tabelle federate.

Limitazioni delle viste materializzate automatizzate

Di seguito sono riportate le limitazioni per l'utilizzo delle viste materializzate automatizzate:

- Numero massimo di AutoMV: il limite delle viste materializzate automatizzate è di 200 per database nel cluster.
- Spazio di archiviazione e capacità: una caratteristica importante di AutoMV è che viene eseguito utilizzando cicli in background di riserva per contribuire a fare in modo che i carichi di lavoro degli utenti non subiscano un impatto negativo. Se il cluster è occupato o sta esaurendo lo spazio di archiviazione, AutoMV interrompe la sua attività. In particolare, all'80% di capacità totale del cluster, non vengono create nuove viste materializzate automatizzate. Al 90% di capacità totale possono essere eliminate per facilitare l'esecuzione dei carichi di lavoro degli utenti senza un peggioramento delle prestazioni. Per ulteriori informazioni sulla determinazione della capacità dei cluster, consulta STV_NODE_STORAGE_CAPACITY.

Fatturazione per le viste materializzate automatizzate

La funzionalità di ottimizzazione automatica di Amazon Redshift crea e aggiorna le viste materializzate automatizzate. Non sono previsti costi per le risorse di calcolo per questo processo. L'archiviazione delle viste materializzate automatizzate viene addebitata alla normale tariffa di archiviazione. Per ulteriori informazioni sui prezzi, consultare Prezzi di Amazon Redshift.

Risorse aggiuntive

Il seguente post sul blog fornisce ulteriori informazioni sulle viste materializzate automatizzate. Descrive nel dettaglio come vengono create, gestite ed eliminate. Spiega anche gli algoritmi alla base di queste decisioni: Optimize your Amazon Redshift query performance with automated materialized views (Ottimizzazione delle prestazioni delle query su Amazon Redshift con viste materializzate automatizzate).

Questo video inizia con una spiegazione delle viste materializzate e mostra come migliorano le prestazioni e si risparmiano risorse. Fornisce quindi una spiegazione approfondita delle viste materializzate automatizzate con un'animazione del flusso di processo e una dimostrazione dal vivo.

Utilizzo di una funzione definita dall'utente (UDF) in una vista materializzata

Puoi utilizzare una UDF scalare in una vista materializzata di Amazon Redshift. Definirle in Python o SQL e fare riferimento ad esse nella definizione della vista materializzata.

Fare riferimento a una UDF in una vista materializzata

La procedura seguente mostra come utilizzare le UDF che eseguono semplici confronti aritmetici, in una definizione di vista materializzata.

1. Creare una tabella da utilizzare nella definizione della vista materializzata.

   ```
   CREATE TABLE base_table (a int, b int);
   ```

2. Creare una funzione scalare definita dall'utente in Python che restituisca un valore booleano che indica se un numero intero è più grande di un numero intero di confronto.

   ```
   CREATE OR REPLACE FUNCTION udf_python_bool(x1 int, x2 int) RETURNS bool IMMUTABLE
   AS $$
     return x1 > x2
   $$ LANGUAGE plpythonu;
   ```

   Facoltativamente, creare una UDF funzionalmente simile con SQL, che è possibile usare per confrontare i risultati con la prima.

   ```
   CREATE OR REPLACE FUNCTION udf_sql_bool(int, int) RETURNS bool IMMUTABLE
   AS $$
     select $1 > $2;
   $$ LANGUAGE SQL;
   ```

3. Creare una vista materializzata che selezioni dalla tabella creata e faccia riferimento alla UDF.

   ```
   CREATE MATERIALIZED VIEW mv_python_udf AS SELECT udf_python_bool(a, b) AS a FROM
    base_table;
   ```

   Facoltativamente, è possibile creare una vista materializzata che fa riferimento alla UDF SQL.

```
CREATE MATERIALIZED VIEW mv_sql_udf AS SELECT udf_sql_bool(a, b) AS a FROM
 base_table;
```

4. Aggiungere i dati alla tabella e aggiornare la vista materializzata.

```
INSERT INTO base_table VALUES (1,2), (1,3), (4,2);
```

```
REFRESH MATERIALIZED VIEW mv_python_udf;
```

Facoltativamente, è possibile creare una vista materializzata che fa riferimento alla UDF SQL.

```
REFRESH MATERIALIZED VIEW mv_sql_udf;
```

5. Eseguire una query sui dati dalla vista materializzata.

```
SELECT * FROM mv_python_udf ORDER BY a;
```

I risultati della query sono i seguenti:

```
a
-----
false
false
true
```

Ciò restituisce `true` per l'ultimo set di valori perché il valore della colonna a (4) è maggiore del valore della colonna b (2).

6. Facoltativamente, puoi eseguire query su una vista materializzata che fa riferimento alla UDF SQL. I risultati della funzione SQL corrispondono ai risultati della versione Python.

```
SELECT * FROM mv_sql_udf ORDER BY a;
```

I risultati della query sono i seguenti:

```
a
-----
false
false
```

```
true
```

Ciò restituisce `true` per l'ultimo set di valori da confrontare.

7. Utilizzare un'istruzione DROP con CASCADE per eliminare la funzione definita dall'utente e la vista materializzata che vi fa riferimento.

```
DROP FUNCTION udf_python_bool(int, int) CASCADE;
```

```
DROP FUNCTION udf_sql_bool(int, int) CASCADE;
```

Importazione di dati in streaming

L'importazione dati in streaming a bassa latenza e ad alta velocità viene eseguita da flusso di dati Amazon Kinesis e da Streaming gestito da Amazon per Apache Kafka in una vista materializzata di Amazon Redshift o di Amazon Redshift serverless con provisioning. Questa operazione riduce il tempo necessario per accedere ai dati e aiuta a diminuire i costi di archiviazione. È possibile configurare l'importazione dati in streaming per il cluster Amazon Redshift o per Amazon Redshift serverless e creare una vista materializzata utilizzando istruzioni SQL, come descritto in Creazione di viste materializzate in Amazon Redshift. Dopodiché, utilizzando l'aggiornamento delle viste materializzate, puoi importare centinaia di megabyte di dati al secondo. Ciò si traduce in un rapido accesso ai dati esterni che vengono rapidamente aggiornati.

Flusso di dati

Un cluster Amazon Redshift con provisioning o un gruppo di lavoro Amazon Redshift serverless è il consumer del flusso. Una vista materializzata è l'area di destinazione dei dati letti dal flusso, che vengono elaborati man mano che arrivano. Ad esempio, i valori JSON possono essere consumati e mappati alle colonne di dati della vista materializzata utilizzando comuni istruzioni SQL. Quando la vista materializzata viene aggiornata, Redshift utilizza i dati dagli shard di dati Kinesis allocati o dalle partizioni Kafka finché la vista non raggiunge la parità con quella per lo stream Kinesis o last per l'argomento Kafka. `SEQUENCE_NUMBER` `Offset` La vista materializzata successiva aggiorna i dati di lettura dall'ultima `SEQUENCE_NUMBER` del precedente aggiornamento fino a quando non raggiunge la parità con i dati del flusso o dell'argomento.

Casi d'uso dell'importazione dati in streaming

I casi d'uso per l'importazione dati in streaming di Amazon Redshift richiedono la gestione di dati generati continuamente (in streaming) e che devono essere elaborati entro un breve periodo (latenza) dalla loro generazione. Si chiama analisi dei dati quasi in tempo reale. Le fonti di dati possono variare e includono dispositivi IOT, dati telemetrici di sistema o dati di clickstream di applicazioni o siti Web occupati.

Considerazioni sull'importazione dati in streaming

Di seguito sono riportate importanti considerazioni e best practice riguardanti le prestazioni e la fatturazione durante la configurazione dell'ambiente di importazione dei dati in streaming.

- Utilizzo e attivazione dell'aggiornamento automatico: le query di aggiornamento automatico per una o più viste materializzate vengono trattate come qualsiasi altro carico di lavoro dell'utente. L'aggiornamento automatico carica i dati dal flusso non appena arrivano.

  L'aggiornamento automatico può essere attivato in modo esplicito per una vista materializzata creata per l'importazione di dati in streaming. A tale scopo, specifica `AUTO REFRESH` nella definizione della vista materializzata. L'impostazione predefinita è l'aggiornamento manuale. Per specificare l'aggiornamento automatico di una vista materializzata esistente per l'importazione di dati in streaming, puoi eseguire `ALTER MATERIALIZED VIEW` per attivarlo. Per ulteriori informazioni, consulta CREATE MATERIALIZED VIEW o ALTER MATERIALIZED VIEW.
- Importazione dati in streaming e Amazon Redshift serverless: le stesse istruzioni per l'impostazione e la configurazione che si applicano all'importazione dati in streaming di Amazon Redshift su un cluster con provisioning si applicano anche all'importazione dati in streaming su Amazon Redshift serverless. È importante dimensionare Amazon Redshift serverless con il livello di RPU necessario per supportare l'importazione dati in streaming con aggiornamento automatico e altri carichi di lavoro. Per ulteriori informazioni consulta Fatturazione per Amazon Redshift Serverless.
- Amazon Redshift nodes in a different availability zone than the Amazon MSK cluster (Nodi Amazon Redshift in una zona di disponibilità diversa rispetto al cluster Amazon MSK): quando si configura l'importazione dati in streaming, Amazon Redshift tenta di connettersi a un cluster Amazon MSK nella stessa zona di disponibilità, se è abilitato il riconoscimento del rack per Amazon MSK. Se tutti i nodi si trovano in zone di disponibilità diverse da quelle del cluster Amazon Redshift, potrebbero venire applicati costi di trasferimento dei dati tra le zone di disponibilità. Per evitare ciò, mantieni almeno un nodo del cluster di broker Amazon MSK nella stessa zona del cluster o del gruppo di lavoro con provisioning Redshift.

- Posizione dell'aggiornamento iniziale: dopo aver creato una vista materializzata, l'aggiornamento iniziale inizia dal `TRIM_HORIZON` di un flusso Kinesis o dall'offset 0 di un argomento Amazon MSK.
- Formati di dati: sono supportati solo i formati di dati che possono essere convertiti da `VARBYTE`. Per ulteriori informazioni, consulta Tipo VARBYTE e Operatori VARBYTE.
- Aggiunta di record a una tabella: puoi eseguire `ALTER TABLE APPEND` per aggiungere righe a una tabella di destinazione da una vista materializzata di origine esistente. Funziona solo se la vista materializzata è configurata per l'importazione dati in streaming. Per ulteriori informazioni, consulta ALTER TABLE APPEND.
- Esecuzione di TRUNCATE o DELETE: puoi rimuovere i record da una vista materializzata utilizzata per l'importazione dati in streaming, utilizzando due metodi:
  - `TRUNCATE`: questo comando elimina tutte le righe da una vista materializzata configurata per l'importazione dati in streaming. Non esegue una scansione della tabella. Per ulteriori informazioni, consulta TRUNCATE.
  - `DELETE`: questo comando elimina tutte le righe da una vista materializzata configurata per l'importazione dati in streaming. Per ulteriori informazioni, consulta DELETE.

Buone pratiche e consigli per l'inserimento dello streaming

In alcuni casi ti vengono presentate delle opzioni su come configurare l'ingestione dello streaming. Consigliamo le seguenti best practice. Queste si basano sui nostri test e aiutano i clienti a evitare problemi che portano alla perdita di dati.

- Estrazione di valori dai dati in streaming: se utilizzi la funzione JSON_EXTRACT_PATH_TEXT nella definizione della vista materializzata per eliminare lo streaming JSON in entrata, ciò può influire in modo significativo su prestazioni e latenza. Per spiegare, per ogni colonna estratta utilizzando JSON_EXTRACT_PATH_TEXT, il JSON in entrata viene analizzato nuovamente. Successivamente, si verifica qualsiasi conversione, filtraggio e logica aziendale del tipo di dati. Ciò significa, ad esempio, che se estrai 10 colonne dai tuoi dati JSON, ogni record JSON viene analizzato 10 volte, il che include conversioni di tipo e logica aggiuntiva. Ciò si traduce in una maggiore latenza di ingestione. Un approccio alternativo che consigliamo consiste nell'utilizzare la funzione JSON_PARSE per convertire i record JSON nel tipo di dati SUPER di Redshift. Dopo che i dati in streaming arrivano nella vista materializzata, usa PartiQL per estrarre singole stringhe dalla rappresentazione dei dati JSON fornita da SUPER. Per ulteriori informazioni, consulta Interrogazione di dati semistrutturati.

È anche importante notare che JSON_EXTRACT_PATH_TEXT ha una dimensione massima dei dati di 64 KB. Pertanto, se un record JSON è più grande di 64 KB, l'elaborazione con JSON_EXTRACT_PATH_TEXT genera un errore.

- Mappatura di uno Amazon Kinesis Data Streams stream o di un argomento Amazon MSK a una vista materializzata di streaming-ingestion di Amazon Redshift: non è consigliabile creare più viste materializzate di streaming-ingestion per importare dati da un singolo flusso o argomento Amazon MSK. Amazon Kinesis Data Streams Questo perché ogni vista materializzata crea un consumatore per ogni shard nel flusso o nella partizione Kinesis Data Streams nell'argomento Kafka. Ciò può comportare una limitazione o un superamento della velocità effettiva dello stream o dell'argomento. Inoltre, può comportare costi più elevati, poiché si acquisiscono gli stessi dati più volte. Ti consigliamo di creare una visualizzazione materializzata in streaming per ogni stream o argomento.

  Se il tuo caso d'uso richiede l'inserimento dei dati da un flusso KDS o da un argomento MSK in più viste materializzate, consulta il blog AWS Big Data, in particolare sulle migliori pratiche per implementare l' near-real-time analisi usando Amazon Redshift Streaming Ingestion with Amazon MSK, prima di farlo.

Utilizzo dell'importazione dati in streaming rispetto ai dati di gestione temporanea in Amazon S3

Esistono diverse opzioni per i dati in streaming su Amazon Redshift o su Amazon Redshift serverless. Due opzioni ben note sono l'ingestione dello streaming, come descritto in questo argomento, o la configurazione di un flusso di distribuzione su Amazon S3 con Firehose. L'elenco seguente descrive ciascun metodo:

1. L'importazione dati in streaming dal flusso di dati Amazon Kinesis o Streaming gestito da Amazon per Apache Kafka ad Amazon Redshift o ad Amazon Redshift serverless implica la configurazione di una vista materializzata per ricevere i dati.
2. La distribuzione di dati in Amazon Redshift tramite Kinesis Data Streams e lo streaming tramite Firehose implica la connessione del flusso di origine ad Amazon Data Firehose e l'attesa che Firehose esegua lo staging dei dati in Amazon S3. Questo processo utilizza batch di varie dimensioni a intervalli di buffer di lunghezza variabile. Dopo lo streaming su Amazon S3, Firehose avvia un comando COPY per caricare i dati.

Con l'importazione dati in streaming, si evitano diverse fasi necessarie per il secondo processo:

- Non è necessario inviare dati a un flusso di distribuzione di Amazon Data Firehose, perché con lo streaming ingestion, i dati possono essere inviati direttamente da Kinesis Data Streams a una vista materializzata in un database Redshift.
- Non è necessario trasferire i dati in streaming in Amazon S3, perché i dati dell'importazione dati in streaming vengono inviati direttamente alla vista materializzata di Redshift.
- Non è necessario scrivere ed eseguire i comandi COPY perché i dati nella vista materializzata vengono aggiornati direttamente dal flusso. Il caricamento dei dati da Amazon S3 a Redshift non fa parte del processo.

Tieni presente che l'importazione dati in streaming è limitata ai flussi dal flusso di dati Amazon Kinesis e agli argomenti di Amazon MSK. Per lo streaming da Kinesis Data Streams verso destinazioni diverse da Amazon Redshift, è probabile che sia necessario un flusso di distribuzione Firehose. Per ulteriori informazioni, consulta Invio di dati a un flusso di distribuzione Amazon Data Firehose.

Considerazioni

Di seguito sono riportate alcune considerazioni per l'inserimento dello streaming in Amazon Redshift.

Caratteristica o comportamento	Descrizione
Limite di lunghezza dell'argomento Kafka	Non è possibile utilizzare un nome di un argomento Kafka di lunghezza superiore a 128 caratteri (escluse le virgolette). Per ulteriori informazioni, consulta Nomi e identificatori.
Aggiornamenti e JOIN incrementali su una vista materializzata	La vista materializzata deve essere gestibile in modo incrementale. Non è possibile eseguire il ricalcolo completo per Kinesis o Amazon MSK perché per impostazione predefinita non conservano la cronologia dei flussi o degli argomenti per più di 24 ore o 7 giorni. È possibile impostare periodi di conservazione dei dati più lunghi in Kinesis o Amazon MSK. Tuttavia, ciò può comportare una manutenzione maggiore e costi superiori. Inoltre, al momento i JOIN non sono supportati nelle viste materializzate create su un flusso Kinesis o su un argomento Amazon MSK. Dopo aver creato una vista materializzata su un flusso o un argomento, è possibile creare un'altra vista materiali

Caratteristica o comportamento	Descrizione
	zzata da utilizzare per connettere la vista materializzata in streaming ad altre viste materializzate, tabelle o viste. Per ulteriori informazioni, consulta REFRESH MATERIALIZED VIEW.
Analisi dei record	L'importazione dati in streaming di Amazon Redshift non supporta l'analisi di record aggregati dalla Kinesis Producer Library (Concetti chiave KPL – Aggregazione). I record aggregati vengono importati, ma vengono archiviati come dati buffer del protocollo binario. (Consulta la sezione relativa ai Buffer di protocollo per ulteriori informazioni). A seconda di come si esegue il push dei dati su Kinesis, potrebbe essere necessario disabilitare questa funzionalità.
Decompressione	Al momento VARBYTE non supporta alcun metodo di decompressione. Pertanto, non è possibile eseguire query sui record contenenti dati compressi in Redshift. Decomprimi i dati prima di eseguirne il push nel flusso Kinesis Streams o nell'argomento Amazon MSK.

Caratteristica o comportamento	Descrizione
Dimensione massima dei record	La dimensione massima dei campi dei record che Amazon Redshift è in grado di importare da Kinesis o Amazon MSK è di poco inferiore a 1 MB. I seguenti punti descrivono in dettaglio il comportamento: • Lunghezza massima VARBYTE: per l'ingestione dello streaming, il `VARBYTE` tipo supporta dati fino a una lunghezza massima di 1.024.000 byte. Kinesis limita i payload a 1 MB. • Limiti dei messaggi: la configurazione predefinita di Amazon MSK limita i messaggi a 1 MB. Inoltre, se un messaggio include intestazioni, la quantità di dati è limitata a 1.048.470 byte. Le impostazioni predefinite non creano problemi di importazione. Tuttavia, è possibile modificare la dimension e massima dei messaggi per Kafka, e quindi per Amazon MSK, impostandola su un valore maggiore. In questo caso, è possibile che il campo chiave/valore di un record Kafka, o l'intestazione, superi il limite di dimensione. Questi record possono causare un errore e non vengono importati. **Note** Amazon Redshift supporta una dimensione massima di 1.024.000 byte per l'acquisizione di streaming da Kinesis o Amazon MSK, anche se Amazon Redshift supporta una dimensione massima di 16 MB per il tipo di dati. `VARBYTE`

Caratteristica o comportamento	Descrizione
Record che generano errori	I record che non possono essere importati in Redshift perché la dimensione dei dati supera quella consentita vengono ignorati. In questo caso, l'aggiornamento della vista materializzata ha comunque esito positivo e un segmento di ogni record che genera un errore viene scritto nella tabella di sistema SYS_STREAM_SCAN_ERRORS. Gli errori derivanti da logica di business, ad esempio un errore in un calcolo o un errore dovuto alla conversione di un tipo, non vengono ignorati. Per evitarli, testa attentamente la logica prima di aggiungerla alla definizione della vista materializzata.
Connettività privata multi-VPC di Amazon MSK	La connettività privata multi-VPC di Amazon MSK non è attualmente supportata per l'ingestione dello streaming Redshift. In alternativa, puoi utilizzare il peering VPC per connettere VPC o AWS Transit Gatewayper connettere VPC e reti locali tramite un hub centrale. Entrambi possono consentire a Redshift di comunicare con un cluster Amazon MSK o con Amazon MSK Serverless in un altro VPC.

Nozioni di base sull'importazione dati in streaming da Amazon Kinesis Data Streams

L'impostazione dell'importazione dati in streaming di Amazon Redshift comporta la creazione di uno schema esterno che mappi all'origine dati in streaming e la creazione di una vista materializzata che faccia riferimento allo schema esterno. L'importazione dati in streaming di Amazon Redshift supporta Kinesis Data Streams come fonte. Pertanto, prima di configurare l'importazione dati in streaming, assicurati che sia presente una fonte di Amazon Kinesis Data Streams. Se non disponi di una fonte, segui le istruzioni nella documentazione di Kinesis in Getting Started with Amazon Kinesis Data Streams o creane una sulla console utilizzando le istruzioni in Creazione di uno stream tramite la console di gestione. AWS

L'importazione dati in streaming di Amazon Redshift utilizza una vista materializzata, che viene aggiornata direttamente dal flusso quando `REFRESH` viene eseguito. La vista materializzata viene

mappata all'origine dati del flusso. È possibile eseguire filtri e aggregazioni sui dati del flusso come parte della definizione della vista materializzata. La vista materializzata dell'importazione dati in streaming (la vista materializzata di base) può fare riferimento a un solo flusso, ma è possibile creare viste materializzate aggiuntive che si uniscono alla vista materializzata di base e con altre viste materializzate o tabelle.

> **Note**
>
> Importazione di dati in streaming e Amazon Redshift Serverless: la procedura di configurazione descritta in questo argomento si applica sia ai cluster con provisioning di Amazon Redshift che ad Amazon Redshift Serverless. Per ulteriori informazioni, consulta Considerazioni sull'importazione dati in streaming.

Supponendo che tu disponga di un flusso Kinesis Data Streams disponibile, il primo passo è definire uno schema in Amazon Redshift con `CREATE EXTERNAL SCHEMA` e fare riferimento a una risorsa Kinesis Data Streams. Successivamente, per accedere ai dati nel flusso, definisci `STREAM` in una vista materializzata. È possibile archiviare i record di flusso in formato `SUPER` semi-strutturato o definire uno schema che comporta la conversione dei dati in tipi di dati Redshift. Quando esegui una query sulla vista materializzata, i record restituiti sono una point-in-time visualizzazione del flusso.

1. Crea un ruolo IAM con una policy di attendibilità che consenta al cluster di Amazon Redshift o al gruppo di lavoro Amazon Redshift serverless di assumere tale ruolo. Per informazioni su come configurare la policy di fiducia per il ruolo IAM, consulta Autorizzazione di Amazon Redshift ad accedere ad AWS altri servizi per tuo conto. Una volta creato, il ruolo dovrebbe avere la seguente policy IAM, che fornisce l'autorizzazione per la comunicazione con il flusso di dati Amazon Kinesis.

   Policy IAM per un flusso non crittografato da Flusso di dati Kinesis

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "ReadStream",
               "Effect": "Allow",
               "Action": [
                   "kinesis:DescribeStreamSummary",
                   "kinesis:GetShardIterator",
   ```

```
                "kinesis:GetRecords",
                "kinesis:DescribeStream"
            ],
            "Resource": "arn:aws:kinesis:*:0123456789:stream/*"
        },
        {
            "Sid": "ListStream",
            "Effect": "Allow",
            "Action": [
                "kinesis:ListStreams",
                "kinesis:ListShards"
            ],
            "Resource": "*"
        }
    ]
}
```

Policy IAM per un flusso crittografato da Flusso di dati Kinesis

```
{
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "ReadStream",
   "Effect": "Allow",
   "Action": [
    "kinesis:DescribeStreamSummary",
    "kinesis:GetShardIterator",
    "kinesis:GetRecords",
    "kinesis:DescribeStream"
   ],
   "Resource": "arn:aws:kinesis:*:0123456789:stream/*"
  },
  {
   "Sid": "DecryptStream",
   "Effect": "Allow",
   "Action": [
    "kms:Decrypt"
   ],
   "Resource": "arn:aws:kms:us-
east-1:0123456789:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
        {
   "Sid": "ListStream",
```

```
    "Effect": "Allow",
    "Action": [
     "kinesis:ListStreams",
     "kinesis:ListShards"
    ],
    "Resource": "*"
   }
 ]
}
```

2. Controlla il tuo VPC e verifica che il cluster Amazon Redshift o Amazon Redshift serverless disponga di un percorso per raggiungere gli endpoint di flusso di dati Kinesis su Internet utilizzando un gateway NAT o Internet. Se desideri che il traffico tra Redshift e Kinesis Data Streams rimanga all'interno della rete, prendi in considerazione AWS l'utilizzo di un endpoint VPC con interfaccia Kinesis. Per ulteriori informazioni, consulta Utilizzo di Amazon Kinesis Data Streams con endpoint VPC di interfaccia.
3. In Amazon Redshift, crea uno schema esterno per mappare i dati da Kinesis a uno schema.

```
CREATE EXTERNAL SCHEMA kds
FROM KINESIS
IAM_ROLE { default | 'iam-role-arn' };
```

L'importazione di dati in streaming per il flusso di dati Amazon Kinesis non richiede un tipo di autenticazione. Utilizza il ruolo IAM definito nell'istruzione `CREATE EXTERNAL SCHEMA` per effettuare richieste al flusso di dati Amazon Kinesis.

Facoltativo: utilizza la parola chiave REGION per specificare la regione in cui Amazon Kinesis Data Streams risiede lo stream MSK o Amazon.

```
CREATE EXTERNAL SCHEMA kds
FROM KINESIS
REGION 'us-west-2'
IAM_ROLE { default | 'iam-role-arn' };
```

In questo esempio, la regione specifica la posizione del flusso di origine. IAM_ROLE è un esempio.

4. Crea una vista materializzata per consumare i dati del flusso. Con un'istruzione come la seguente, se un record non può essere analizzato, viene generato un errore. Usa un comando come questo se non vuoi che i record di errore vengano ignorati.

```
CREATE MATERIALIZED VIEW my_view AUTO REFRESH YES AS
SELECT *
FROM kds.my_stream_name;
```

L'esempio seguente definisce una vista materializzata per i dati di origine in formato JSON. La vista verifica che i dati in entrata siano formattati correttamente in JSON. I nomi del flusso di dati Kinesis fanno distinzione tra maiuscole e minuscole e possono contenere lettere maiuscole e minuscole. Per importare da stream con nomi in maiuscolo, puoi impostare la configurazione a livello di database. `enable_case_sensitive_identifier true` Per ulteriori informazioni, consulta Nomi e identificatori e enable_case_sensitive_identifier.

```
CREATE MATERIALIZED VIEW my_view AUTO REFRESH YES AS
SELECT approximate_arrival_timestamp,
partition_key,
shard_id,
sequence_number,
refresh_time,
JSON_PARSE(kinesis_data) as kinesis_data
FROM kds.my_stream_name
WHERE CAN_JSON_PARSE(kinesis_data);
```

Per attivare l'aggiornamento automatico, usa `AUTO REFRESH YES`. Il comportamento predefinito prevede l'aggiornamento manuale. Nota che quando usi CAN_JSON_PARSE, è possibile che i record che non possono essere analizzati vengano ignorati.

Le colonne di metadati includono quanto segue:

Colonna di metadati	Tipo di dati	Descrizione
approximate_arrival_timestamp	timestamp without time zone	L'ora approssimativa in cui il record è stato inserito nel flusso Kinesis
partition_key	varchar(256)	La chiave utilizzata da Kinesis per assegnare il record a una partizione

Colonna di metadati	Tipo di dati	Descrizione
shard_id	char(20)	L'identificatore univoco della partizione all'interno del flusso da cui è stato recuperato il record
sequence_number	varchar(128)	L'identificatore univoco del record dalla partizione Kinesis
ora di aggiornamento	timestamp without time zone	L'ora di inizio dell'aggi ornamento.
kinesis_data	varbyte	Il record dal flusso Kinesis

È importante notare che, se nella definizione della vista materializzata è inclusa una logica aziendale, che in alcuni casi gli errori di logica aziendale possono causare il blocco dell'ingestione dello streaming. Ciò potrebbe comportare la necessità di abbandonare e ricreare la vista materializzata. Per evitare ciò, ti consigliamo di mantenere la logica il più semplice possibile e di eseguire la maggior parte dei controlli di logica aziendale sui dati dopo che sono stati inseriti.

5. Aggiorna la vista per richiamare Redshift ed eseguire la lettura dal flusso e il caricamento dei dati nella vista materializzata.

```
REFRESH MATERIALIZED VIEW my_view;
```

6. Esegui una query sui dati nella vista materializzata.

```
select * from my_view;
```

Nozioni di base sull'importazione dati in streaming da Amazon Managed Streaming per Apache Kafka

Lo scopo dell'importazione dati in streaming di Amazon Redshift è semplificare il processo di importazione diretta di dati di flusso da un servizio di streaming in Amazon Redshift o in Amazon

Redshift serverless. È compatibile con Amazon MSK, Amazon MSK serverless e Kinesis. L'importazione dati in streaming di Amazon Redshift elimina la necessità di gestire in via temporanea un flusso di dati Kinesis o un argomento Amazon MSK in Amazon S3 prima di importare i dati di flusso in Redshift.

A un livello tecnico, l'importazione dati in streaming sia da Amazon Kinesis Data Streams che da Amazon Managed Streaming per Apache Kafka consente l'importazione a bassa latenza e ad alta velocità dei dati di un flusso o di un argomento in una vista materializzata di Amazon Redshift. Dopo la configurazione, utilizzando l'aggiornamento della vista materializzata è possibile acquisire grandi volumi di dati.

Configura l'importazione dati in streaming di Amazon Redshift per Amazon MSK attenendoti alla seguente procedura:

1. Crea uno schema esterno mappato all'origine dati di flusso.
2. Crea una vista materializzata che faccia riferimento allo schema esterno.

Devi disporre di un'origine Amazon MSK prima di configurare l'importazione dati in streaming di Amazon Redshift. Se non disponi di un'origine, segui le istruzioni riportate nella Guida introduttiva all'uso di Amazon MSK.

> **Note**
>
> Importazione di dati in streaming e Amazon Redshift Serverless: la procedura di configurazione descritta in questo argomento si applica sia ai cluster con provisioning di Amazon Redshift che ad Amazon Redshift Serverless. Per ulteriori informazioni, consulta Considerazioni sull'importazione dati in streaming.

Configurazione di IAM ed esecuzione dell'importazione dati in streaming da Kafka

Supponendo che abbia a disposizione un cluster Amazon MSK, il primo passo è definire uno schema in Redshift con `CREATE EXTERNAL SCHEMA` e fare riferimento all'argomento Kafka come origine dati. Quindi, definisci lo `STREAM` in una vista materializzata per accedere ai dati nell'argomento. È possibile archiviare i record in formato `SUPER` semi-strutturato o definire uno schema che comporta la conversione dei dati in tipi di dati Amazon Redshift. Quando si esegue una query sulla vista materializzata, i record restituiti sono una point-in-time visualizzazione dell'argomento.

1. Crea un ruolo IAM con una policy di attendibilità che consenta al cluster di Amazon Redshift o ad Amazon Redshift serverless di assumere tale ruolo. Per informazioni su come configurare la policy di fiducia per il ruolo IAM, consulta Autorizzazione di Amazon Redshift ad accedere ad AWS altri servizi per tuo conto. Una volta creato, al ruolo dovrebbe essere applicata la seguente policy IAM, che fornisce l'autorizzazione per la comunicazione con il cluster Amazon MSK. La policy necessaria dipende dal metodo di autenticazione utilizzato nel cluster, se si utilizza Amazon MSK. Vedi Autenticazione e autorizzazione per le API Apache Kafka per i metodi di autenticazione disponibili in Amazon MSK.

   Una policy IAM per Amazon MSK quando si utilizza l'accesso non autenticato:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "kafka:GetBootstrapBrokers"
            ],
            "Resource": "*"
        }
    ]
}
```

   Una policy IAM per Amazon MSK quando si utilizza l'autenticazione IAM:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "MSKIAMpolicy",
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeTopic",
                "kafka-cluster:Connect"
            ],
            "Resource": [
                "arn:aws:kafka:*:0123456789:cluster/MyTestCluster/*",
                "arn:aws:kafka:*:0123456789:topic/MyTestCluster/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:*:0123456789:group/MyTestCluster/*"
            ]
        },
        {
            "Sid": "MSKPolicy",
            "Effect": "Allow",
            "Action": [
                "kafka:GetBootstrapBrokers"
            ],
            "Resource": "*"
        }
    ]
}
```

2. Controlla il tuo VPC e verifica che il cluster Amazon Redshift o Amazon Redshift serverless disponga di un percorso per raggiungere il cluster Amazon MSK. Le regole in entrata del gruppo di sicurezza per il cluster Amazon MSK dovrebbero consentire il gruppo di sicurezza del cluster Amazon Redshift o il gruppo di sicurezza del gruppo di lavoro Amazon Redshift serverless. Le porte specificate dipendono dal metodo di autenticazione utilizzato per il cluster, se utilizzi Amazon MSK. Per ulteriori informazioni, consulta Informazioni sulle porte e Accesso dall'interno AWS ma dall'esterno del VPC.

   Notare che l'autenticazione client con mTLS non è supportata per l'importazione di dati in streaming. Per ulteriori informazioni, consulta Limitazioni.

   La tabella seguente mostra le opzioni di configurazione aggiuntive da impostare per l'acquisizione di streaming da Amazon MSK:

Configurazione di Amazon Redshift	Configurazione di Amazon MQ	Porta da aprire tra Redshift e Amazon MSK
AUTHENTICATION NONE	Trasporto TLS disabilitato	9092

Configurazione di Amazon Redshift	Configurazione di Amazon MQ	Porta da aprire tra Redshift e Amazon MSK
AUTHENTICATION NONE	Trasporto TLS abilitato	9094
AUTHENTICATION IAM	IAM	9098/9198

L'autenticazione Amazon Redshift è impostata nell'istruzione CREATE EXTERNAL SCHEMA.

Nel caso in cui il cluster Amazon MSK ha l'autenticazione Mutual Transport Layer Security (mTLS) abilitata, la configurazione di Amazon Redshift per l'utilizzo di AUTHENTICATION NONE indica ad Amazon Redshift di utilizzare la porta 9094 per l'accesso non autenticato. Tuttavia, poiché la porta viene utilizzata dall'autenticazione mTLS, questa operazione avrà esito negativo. Per questo motivo, consigliamo di passare a AUTHENTICATION IAM quando utilizzi mTLS.

3. Abilita il routing VPC avanzato nel cluster Amazon Redshift. o nel gruppo di lavoro Amazon Redshift serverless. Per ulteriori informazioni, consulta Abilitazione del routing VPC avanzato.

> **Note**
>
> Per recuperare l'URL del broker di bootstrap di Amazon MSK, Amazon Redshift effettua una chiamata all'API GetBootstrapBrokers, utilizzando le autorizzazioni fornite dal ruolo IAM associato. Tieni presente che, affinché questa richiesta abbia esito positivo quando è abilitato il routing VPC avanzato, la sottorete del cluster con provisioning di Amazon Redshift o del gruppo di lavoro Serverless Amazon Redshift deve disporre di un gateway NAT o un gateway Internet. Gli ACL di rete e le regole in uscita dei gruppi di sicurezza per la suddetta sottorete devono inoltre consentire l'accesso agli endpoint del servizio API Amazon MSK. Per ulteriori informazioni, consulta Amazon Managed Streaming for Apache Kafka endpoint and quote.

4. In Amazon Redshift, crea uno schema esterno da mappare al cluster Amazon MSK.

```
CREATE EXTERNAL SCHEMA MySchema
FROM MSK
IAM_ROLE { default | 'iam-role-arn' }
AUTHENTICATION { none | iam }
CLUSTER_ARN 'msk-cluster-arn';
```

Nella clausola FROM, Amazon MSK indica che lo schema mappa i dati dai servizi gestiti Kafka.

L'importazione di dati in streaming per Amazon MSK fornisce i seguenti tipi di autenticazione, quando crei lo schema esterno:

- none: specifica che non è presente alcuna procedura di autenticazione.
- iam: specifica l'autenticazione IAM. Quando scegli questa opzione, assicurati che il ruolo IAM disponga delle autorizzazioni per l'autenticazione IAM.

Gli ulteriori metodi di autenticazione di Amazon MSK, come l'autenticazione TLS o nome utente e password, non sono supportati per l'importazione di dati in streaming.

CLUSTER_ARN specifica il cluster Amazon MSK da cui stai eseguendo lo streaming.

5. Crea una vista materializzata per consumare i dati dall'argomento. Usa un comando SQL come questo esempio se non vuoi che i record di errore vengano ignorati.

```
CREATE MATERIALIZED VIEW MyView AUTO REFRESH YES AS
SELECT *
FROM MySchema."mytopic";
```

L'esempio seguente definisce una vista materializzata con i dati sorgente JSON. Nota: la vista seguente verifica che i dati siano di tipo JSON e utf8 validi. I nomi degli argomenti Kafka distinguono tra maiuscole e minuscole e possono contenere lettere maiuscole e minuscole. Per importare da argomenti con nomi in maiuscolo, puoi impostare la configurazione enable_case_sensitive_identifier a true livello di database. Per ulteriori informazioni, consulta Nomi e identificatori e enable_case_sensitive_identifier.

```
CREATE MATERIALIZED VIEW MyView AUTO REFRESH YES AS
SELECT kafka_partition,
 kafka_offset,
 kafka_timestamp_type,
 kafka_timestamp,
 kafka_key,
 JSON_PARSE(kafka_value) as kafka_data,
 kafka_headers,
 refresh_time
FROM MySchema."mytopic"
WHERE CAN_JSON_PARSE(kafka_value);
```

Per attivare l'aggiornamento automatico, usa `AUTO REFRESH YES`. Il comportamento predefinito prevede l'aggiornamento manuale.

Le colonne di metadati includono quanto segue:

Colonna di metadati	Tipo di dati	Descrizione
kafka_partition	bigint	ID della partizione del record dall'argomento Kafka
kafka_offset	bigint	Offset del record nell'argomento di Kafka per una determinata partizione
kafka_timestamp_type	char(1)	Tipo di timestamp utilizzato nel record Kafka: • C: registra l'ora di creazione (CREATE_TIME) sul lato client • L: registra l'ora di aggiunta (LOG_APPEND_TIME) sul lato server Kafka • U: l'ora di creazione del record non è disponibile (NO_TIMESTAMP_TYPE)
kafka_timestamp	timestamp without time zone	Il formato del valore timestamp per il record
kafka_key	varbyte	La chiave del record Kafka
kafka_value	varbyte	Il record ricevuto da Kafka
kafka_headers	super	L'intestazione del record ricevuto da Kafka

Colonna di metadati	Tipo di dati	Descrizione
ora di aggiornamento	timestamp without time zone	L'ora di inizio dell'aggiornamento.

È importante notare che, se nella definizione della vista materializzata è inclusa la logica aziendale, che in alcuni casi gli errori di logica aziendale possono causare un blocco nell'acquisizione dello streaming. Ciò potrebbe comportare la necessità di abbandonare e ricreare la vista materializzata. Per evitare ciò, ti consigliamo di mantenere la logica aziendale semplice e di eseguire una logica aggiuntiva sui dati dopo averli inseriti.

6. Aggiorna la vista per invocare Amazon Redshift ed eseguire la lettura dall'argomento e il caricamento dei dati nella vista materializzata.

```
REFRESH MATERIALIZED VIEW MyView;
```

7. Esegui una query sui dati nella vista materializzata.

```
select * from MyView;
```

La vista materializzata viene aggiornata direttamente dall'argomento quando si esegue il comando REFRESH. Viene creata una vista materializzata che corrisponde all'origine dati dell'argomento Kafka. È possibile eseguire filtri e aggregazioni sui dati nell'ambito della definizione della vista materializzata. La vista materializzata dell'importazione dati in streaming (vista materializzata di base) può fare riferimento a un solo argomento Kafka, ma è possibile creare viste materializzate aggiuntive che si uniscono alla vista materializzata di base e con altre viste materializzate o tabelle.

Per ulteriori informazioni sulle limitazioni relative all'importazione dati in streaming, consulta Considerazioni.

Tutorial sull'importazione dati in streaming delle stazioni dei veicoli elettrici con Kinesis

Questa procedura illustra come acquisire dati da un flusso Kinesis denominato ev_station_data, che contiene dati di consumo provenienti da diverse stazioni di ricarica per veicoli elettrici, in formato

JSON. Lo schema è ben definito. L'esempio mostra come archiviare i dati come JSON grezzi e come convertire i dati JSON in tipi di dati Amazon Redshift man mano che vengono importati.

Configurazione del produttore

1. Utilizzando Amazon Kinesis Data Streams, segui la procedura per creare un flusso denominato `ev_station_data`. Scegli On-demand (On demand) per Capacity mode (Modalità capacità). Per ulteriori informazioni, consulta Creazione di uno stream tramite la console AWS di gestione.
2. Amazon Kinesis Data Generator può aiutarti a generare dati di test da utilizzare con il tuo flusso. Segui i passaggi descritti nello strumento per iniziare e utilizza il seguente modello di dati per la generazione dei dati:

```
{

   "_id" : "{{random.uuid}}",
   "clusterID": "{{random.number(
        {   "min":1,
            "max":50
        }
    )}}",
    "connectionTime": "{{date.now("YYYY-MM-DD HH:mm:ss")}}",
    "kWhDelivered": "{{commerce.price}}",
    "stationID": "{{random.number(
        {   "min":1,
            "max":467
        }
    )}}",
      "spaceID": "{{random.word}}-{{random.number(
        {   "min":1,
            "max":20
        }
    )}}",

   "timezone": "America/Los_Angeles",
   "userID": "{{random.number(
        {   "min":1000,
            "max":500000
        }
    )}}"
}
```

Ogni oggetto JSON nei dati di flusso ha le seguenti proprietà:

```
{
    "_id": "12084f2f-fc41-41fb-a218-8cc1ac6146eb",
    "clusterID": "49",
    "connectionTime": "2022-01-31 13:17:15",
    "kWhDelivered": "74.00",
    "stationID": "421",
    "spaceID": "technologies-2",
    "timezone": "America/Los_Angeles",
    "userID": "482329"
}
```

Configurazione di Amazon Redshift

In questa procedura viene illustrato come configurare la vista materializzata per l'importazione dati.

1. Crea uno schema esterno per mappare i dati da Kinesis a un oggetto Redshift.

   ```
   CREATE EXTERNAL SCHEMA evdata FROM KINESIS
   IAM_ROLE 'arn:aws:iam::0123456789:role/redshift-streaming-role';
   ```

   Per informazioni su come configurare il ruolo IAM, consulta Nozioni di base sull'importazione dati in streaming da Amazon Kinesis Data Streams.

2. Crea una vista materializzata per consumare i dati del flusso. Gli esempi seguenti mostrano entrambi i metodi di definizione delle viste materializzate per l'importazione dati di origine JSON.

   Innanzitutto, archivia i record di flusso in formato SUPER semi-strutturato. In questo esempio, l'origine JSON viene archiviata in Redshift senza convertirsi in tipi Redshift.

   ```
   CREATE MATERIALIZED VIEW ev_station_data AS
       SELECT approximate_arrival_timestamp,
       partition_key,
       shard_id,
       sequence_number,
       json_parse(kinesis_data) as payload
       FROM evdata."ev_station_data" WHERE can_json_parse(kinesis_data);
   ```

Al contrario, nella seguente definizione di vista materializzata, la vista materializzata ha uno schema definito in Redshift. La vista materializzata viene distribuita sul valore UUID dal flusso e viene ordinata in base al valore `approximatearrivaltimestamp`.

```
CREATE MATERIALIZED VIEW ev_station_data_extract DISTKEY(6) sortkey(1) AUTO REFRESH
 YES AS
    SELECT refresh_time,
    approximate_arrival_timestamp,
    partition_key,
    shard_id,
    sequence_number,

 json_extract_path_text(from_varbyte(kinesis_data,'utf-8'),'_id',true)::character(36)
 as ID,

 json_extract_path_text(from_varbyte(kinesis_data,'utf-8'),'clusterID',true)::varchar(30)
 as clusterID,

 json_extract_path_text(from_varbyte(kinesis_data,'utf-8'),'connectionTime',true)::varchar(
 as connectionTime,

 json_extract_path_text(from_varbyte(kinesis_data,'utf-8'),'kWhDelivered',true)::DECIMAL(10
 as kWhDelivered,

 json_extract_path_text(from_varbyte(kinesis_data,'utf-8'),'stationID',true)::DECIMAL(10,2)
 as stationID,

 json_extract_path_text(from_varbyte(kinesis_data,'utf-8'),'spaceID',true)::varchar(100)
 as spaceID,
    json_extract_path_text(from_varbyte(kinesis_data,
 'utf-8'),'timezone',true)::varchar(30)as timezone,

 json_extract_path_text(from_varbyte(kinesis_data,'utf-8'),'userID',true)::varchar(30)
 as userID
    FROM evdata."ev_station_data"
    WHERE LENGTH(kinesis_data) < 65355;
```

Esecuzione di una query sul flusso

1. Esegui una query sulla vista materializzata aggiornata per ottenere statistiche di utilizzo.

```
SELECT to_timestamp(connectionTime, 'YYYY-MM-DD HH24:MI:SS') as connectiontime
,SUM(kWhDelivered) AS Energy_Consumed
,count(distinct userID) AS #Users
from ev_station_data_extract
group by  to_timestamp(connectionTime, 'YYYY-MM-DD HH24:MI:SS')
order by 1 desc;
```

2. Visualizza i risultati.

```
connectiontime         energy_consumed #users
2022-02-08 16:07:21+00   4139         10
2022-02-08 16:07:20+00   5571         10
2022-02-08 16:07:19+00   8697         20
2022-02-08 16:07:18+00   4408         10
2022-02-08 16:07:17+00   4257         10
2022-02-08 16:07:16+00   6861         10
2022-02-08 16:07:15+00   5643         10
2022-02-08 16:07:14+00   3677         10
2022-02-08 16:07:13+00   4673         10
2022-02-08 16:07:11+00   9689         20
```

Creazione di viste nel AWS Glue Data Catalog (anteprima)

Questa è una documentazione di pre-rilascio per le viste del Catalogo dati per Amazon Redshift, che è nella versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche . Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms.

È possibile creare un cluster Amazon Redshift di anteprima per testare nuove funzionalità di Amazon Redshift. Non è possibile utilizzare queste funzionalità in produzione o spostare il cluster di anteprima in un cluster di produzione o in un cluster su un'altra traccia. Per i termini e le condizioni dell'anteprima, consulta la sezione relativa a beta e anteprime nei termini del servizio AWS.

Come creare un cluster di anteprima

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegli Provisioned clusters dashboard (Pannello di controllo dei cluster con provisioning) e seleziona Clusters (Cluster). Regione AWS Sono elencati i cluster del tuo account nella versione corrente. Nelle colonne dell'elenco è visualizzato un sottoinsieme delle proprietà di ciascun cluster.
3. Nella pagina dell'elenco dei cluster viene visualizzato un banner che introduce l'anteprima. Scegli il pulsante Create preview cluster (Crea cluster di anteprima) per aprire la pagina di creazione del cluster.
4. Inserisci le proprietà del cluster. Scegli la traccia di anteprima che contiene le funzionalità che desideri testare. Consigliamo di assegnare al cluster un nome che indichi che si trova in una traccia di anteprima. Scegli le opzioni per il cluster, tra cui quelle contrassegnate come - anteprima, per le funzionalità che desideri testare. Per informazioni generali sulla creazione di cluster, consulta Creazione di un cluster nella Guida alla gestione di Amazon Redshift.
5. Per creare un cluster di anteprima, scegli Crea cluster.

> **Note**
>
> La traccia `preview_2023` è la traccia di anteprima più recente disponibile. Questa traccia supporta la creazione di cluster solo con tipi di nodo RA3. Il tipo di nodo DC2 e qualsiasi tipo di nodo precedente non sono supportati.

6. Quando il cluster di anteprima è disponibile, utilizza il client SQL per caricare dati ed eseguire query su di essi.

La funzionalità di anteprima delle viste del Catalogo dati è disponibile solo nelle regioni riportate di seguito.

- Stati Uniti orientali (Ohio) (us-east-2)
- Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Stati Uniti occidentali (California settentrionale) (us-west-1)
- Asia Pacifico (Tokyo) (ap-northeast-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Stoccolma) (eu-north-1)

Puoi anche creare un gruppo di lavoro di anteprima per testare le viste del catalogo dati. Non è possibile utilizzare queste funzionalità in produzione o trasferire il gruppo di lavoro in un altro gruppo di lavoro. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms. Per istruzioni su come creare un gruppo di lavoro di anteprima, consulta Creazione di un gruppo di lavoro di anteprima.

Creando viste in AWS Glue Data Catalog, puoi creare un unico schema di visualizzazione comune e un oggetto di metadati da utilizzare su più motori come Amazon Athena e Amazon EMR Spark. In questo modo puoi utilizzare le stesse viste su data lake e data warehouse e adattarle ai tuoi casi d'uso. Le viste nel Catalogo dati sono speciali in quanto sono classificate come viste di definizione, in cui le autorizzazioni di accesso sono definite dall'utente che ha creato la vista anziché dall'utente che esegue la query sulla vista. Di seguito sono riportati alcuni casi d'uso e i vantaggi derivanti dalla creazione di viste nel Catalogo dati:

- Crea una vista che limiti l'accesso ai dati in base alle autorizzazioni necessarie all'utente. Ad esempio, puoi utilizzare le viste nel Catalogo dati per impedire ai dipendenti che non lavorano nel reparto delle risorse umane di visualizzare le informazioni di identificazione personale (PII).
- Assicurati che gli utenti non possano accedere ai record incompleti. Applicando i filtri alla vista nel Catalogo dati, ti assicuri che i record di dati all'interno di una vista nel Catalogo dati siano sempre completi.
- Le viste del Catalogo dati includono un vantaggio di sicurezza nel garantire che la definizione della query utilizzata debba essere completata per creare la vista. Questo vantaggio di sicurezza significa che le viste del Catalogo dati non sono suscettibili ai comandi SQL di utenti malintenzionati.
- Le viste del Catalogo dati offrono gli stessi vantaggi delle viste normali, ad esempio consentono agli utenti di accedere a una vista senza rendere disponibile la tabella sottostante.

Per creare una vista nel Catalogo dati è necessario disporre di una tabella esterna Spectrum, un oggetto contenuto in un'unità di condivisione dati gestita da Lake Formation o una tabella Apache Iceberg.

Le definizioni delle viste del Catalogo dati sono archiviate nel AWS Glue Data Catalog. Utilizzalo AWS Lake Formation per concedere l'accesso tramite concessioni di risorse, concessioni di colonne o controlli di accesso basati su tag. Per ulteriori informazioni sull'assegnazione e la revoca dell'accesso a Lake Formation, consulta Granting and revoking permissions on Data Catalog resources.

Prerequisiti

Prima di poter creare una vista nel Catalogo dati, assicurati di aver soddisfatto i seguenti prerequisiti:

- Assicurati che il ruolo IAM abbia la seguente policy di trust.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
               "Service": [
                    "glue.amazonaws.com",
                    "lakeformation.amazonaws.com"
```

```
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

- È inoltre necessaria la seguente policy per il passaggio di ruoli.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1",
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "glue.amazonaws.com",
                        "lakeformation.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

- Infine, sono necessarie anche le seguenti autorizzazioni.
  - `Glue:GetDatabase`
  - `Glue:GetDatabases`
  - `Glue:CreateTable`
  - `Glue:GetTable`
  - `Glue:UpdateTable`
  - `Glue:DeleteTable`
  - `Glue:GetTables`
  - `Glue:SearchTables`

- `Glue:BatchGetPartition`
- `Glue:GetPartitions`
- `Glue:GetPartition`
- `Glue:GetTableVersion`
- `Glue:GetTableVersions`

Un esempio nd-to-end

Inizia creando uno schema esterno basato sul database Catalogo dati.

```
CREATE EXTERNAL SCHEMA IF NOT EXISTS external_schema FROM DATA CATALOG DATABASE
 'external_data_catalog_db'
IAM_ROLE 'arn:aws:iam::123456789012:role/sample-role';
```

Ora puoi creare una vista nel Catalogo dati.

```
CREATE EXTERNAL PROTECTED VIEW external_schema.remote_view
AS SELECT * FROM external_schema.remote_table;
```

Puoi quindi iniziare a eseguire le query sulla tua vista.

```
SELECT * FROM external_schema.remote_view;
```

Per ulteriori informazioni sui comandi SQL relativi alle viste nel Catalogo dati, consulta CREATE EXTERNAL VIEW, ALTER EXTERNAL VIEW e DROP EXTERNAL VIEW.

Considerazioni e limitazioni

Di seguito sono riportate le considerazioni e le limitazioni che si applicano alle viste create nel Catalogo dati.

- Non è possibile creare una vista del Catalogo dati basata su un'altra vista.
- È possibile avere solo 10 tabelle di base in una vista del Catalogo dati.
- Il definitore della vista deve disporre delle autorizzazioni `SELECT GRANTABLE` complete per le tabelle di base.

- Le viste possono contenere solo oggetti e integrazioni di Lake Formation. I seguenti oggetti non sono consentiti all'interno di una vista.
  - Tabelle di sistema
  - Funzioni definite dall'utente (FDU)
  - Tabelle, viste, viste materializzate e viste con associazione tardiva di Redshift che non si trovano in una condivisione dati gestita da Lake Formation.
- Le viste non possono contenere tabelle Redshift Spectrum annidate.
- È possibile eseguire le query sulle viste solo utilizzando la notazione a due punti. L'esecuzione di query sulle viste di Lake Formationviews da un database montato esternamente non è supportata.
- L'ARN di una tabella Lake Formation a cui si fa riferimento in una vista Redshift deve contenere meno di 127 caratteri.
- AWS Glue le rappresentazioni degli oggetti di base di una vista devono trovarsi nella stessa Account AWS regione della vista.

Query su dati spaziali in Amazon Redshift

I dati spaziali descrivono la posizione e la forma di una geometria in uno spazio definito (un sistema di riferimento spaziale). Amazon Redshift supporta i dati spaziali con il tipo di dati GEOMETRY e GEOGRAPHY, che contiene dati spaziali e facoltativamente l'identificatore del sistema di riferimento spaziale (SRID) dei dati.

I dati spaziali contengono dati geometrici che possono essere utilizzati per rappresentare le caratteristiche geografiche. Esempi di questo tipo di dati includono i bollettini meteorologici, le indicazioni stradali sulle mappe, i tweet con posizioni geografiche, le ubicazioni dei negozi e le tratte delle compagnie aeree. I dati spaziali svolgono un ruolo importante per le attività di analisi, di creazione di report e di previsione aziendali.

È possibile eseguire una query sui dati spaziali con le funzioni SQL di Amazon Redshift. i dati spaziali contengono i valori geometrici di un oggetto.

Le operazioni del tipo di dati GEOMETRY funzionano sul piano cartesiano. Sebbene l'identificatore di sistema di riferimento spaziale (SRID) sia memorizzato all'interno dell'oggetto, è semplicemente un identificatore del sistema di coordinate e non svolge alcun ruolo negli algoritmi utilizzati per elaborare gli oggetti GEOMETRY. Al contrario, le operazioni sul tipo di dati GEOGRAPHY utilizzano le coordinate all'interno degli oggetti come coordinate sferiche su uno sferoide. Questo sferoide è definito dallo SRID, che fa riferimento a un sistema di riferimento spaziale geografico. Per impostazione predefinita, i tipi di dati GEOGRAPHY vengono creati con riferimento spaziale (SRID) 4326, che fa riferimento al World Geodetic System (WGS) 84. Per ulteriori informazioni sugli SRID, consulta Spatial reference system in Wikipedia.

È possibile utilizzare la funzione ST_Transform per trasformare le coordinate da diversi sistemi di riferimento spaziale. Al termine della trasformazione delle coordinate, è possibile utilizzare anche un semplice cast tra le due, purché l'input GEOMETRY sia codificato con lo SRID geografico. Questo cast copia semplicemente le coordinate senza ulteriori trasformazioni. Ad esempio:

```
SELECT ST_AsEWKT(ST_GeomFromEWKT('SRID=4326;POINT(10 20)')::geography);
```

```
st_asewkt
------------------------
 SRID=4326;POINT(10 20)
```

Per capire meglio la differenza tra i tipi di dati GEOMETRY e GEOGRAPHY, bisogna prendere in considerazione il calcolo della distanza tra l'aeroporto di Berlino (BER) e l'aeroporto di San Francisco (SFO) utilizzando il World Geodetic System (WGS) 84. Usando il tipo di dati GEOGRAPHY, il risultato è in metri. Quando utilizzi il tipo di dati GEOMETRY con SRID 4326, il risultato è espresso in gradi, che non possono essere convertiti in metri perché la distanza di un grado dipende da dove si trovano le geometrie del globo.

I calcoli sul tipo di dati GEOGRAPHY sono utilizzati principalmente per calcoli realistici di terra rotonda come l'area precisa di un paese senza distorsioni. Ma sono molto più costosi da calcolare. Pertanto, ST_Transform può trasformare le coordinate in un sistema di coordinate locale proiettato appropriato e fare il calcolo sul tipo di dati GEOMETRY più veloce.

Utilizzando i dati spaziali, è possibile eseguire query per i seguenti scopi:

- Individuare la distanza tra due punti;
- Controllare se un'area (poligono) ne contiene un'altra;
- Controllare se una linea interseca un'altra linea o un poligono.

Per contenere i valori dei dati spaziali è possibile usare il tipo di dato GEOMETRY. Un valore GEOMETRY in Amazon Redshift può definire tipi di dati primitivi geometrici bidimensionali (2D), tridimensionali (3DZ), bidimensionali con una misura (3DM) e quadridimensionali (4D):

- Una geometria bidimensionale (2D) è specificata da due coordinate cartesiane (x, y) in un piano.
- Una geometria tridimensionale (3DZ) è specificata da tre coordinate cartesiane (x, y, z) nello spazio.
- Una geometria bidimensionale con misura (3DM) è specificata da tre coordinate (x, y, m), dove le prime due sono coordinate cartesiane in un piano e la terza è una misura.
- Una geometria quadridimensionale (4D) è specificata da quattro coordinate (x, y, z, m), dove le prime tre sono coordinate cartesiane in uno spazio e la quarta è una misura.

Per ulteriori informazioni sui tipi di dati primitivi geometrici, consultare Rappresentazione geometrica tramite Well-known Text su Wikipedia.

Per contenere i valori dei dati spaziali è possibile usare il tipo di dato GEOGRAPHY. Un valore GEOGRAPHY in Amazon Redshift può definire tipi di dati primitivi geometrici bidimensionali (2D), tridimensionali (3DZ), bidimensionali con una misura (3DM) e quadridimensionali (4D):

- Una geometria bidimensionale (2D) è specificata dalle coordinate di longitudine e latitudine su uno sferoide.
- Una geometria tridimensionale (3DZ) è specificata dalle coordinate di longitudine, latitudine e altitudine su uno sferoide.
- Una geometria bidimensionale con misura (3DM) è specificata da tre coordinate (longitudine, latitudine, misura), dove le prime due sono coordinate cartesiane in un piano e la terza è una misura.
- Una geometria quadridimensionale (4D) è specificata da quattro coordinate (longitudine, latitudine, altitudine, misura), dove le prime tre sono longitudine, latitudine e altitudine e la quarta è una misura.

Per ulteriori informazioni sui sistemi di coordinate geografici, consultare Sistema di coordinate geografici e Sistema di coordinate sferico su Wikipedia.

Il tipo di dati `GEOMETRY` e `GEOGRAPHY` presenta i seguenti sottotipi:

- `POINT`
- `LINESTRING`
- `POLYGON`
- `MULTIPOINT`
- `MULTILINESTRING`
- `MULTIPOLYGON`
- `GEOMETRYCOLLECTION`

Sono presenti delle funzioni SQL di Amazon Redshift che supportano le seguenti rappresentazioni dei dati geometrici:

- GeoJSON
- Well-known text (WKT)
- Extended well-known text (EWKT)
- Rappresentazione Well-known binary (WKB)
- Extended well-known binary (EWKB)

È possibile eseguire il casting tra i tipi di dati `GEOMETRY` e `GEOGRAPHY`.

Il seguente codice SQL lancia una linestring da un GEOMETRY a GEOGRAPHY.

```
SELECT ST_AsEWKT(ST_GeomFromText('LINESTRING(110 40, 2 3, -10 80, -7 9)')::geography);
```

```
 st_asewkt
--------------------------------------------
 SRID=4326;LINESTRING(110 40,2 3,-10 80,-7 9)
```

Il seguente codice SQL lancia una linestring da un GEOGRAPHY a GEOMETRY.

```
SELECT ST_AsEWKT(ST_GeogFromText('LINESTRING(110 40, 2 3, -10 80, -7 9)')::geometry);
```

```
 st_asewkt
--------------------------------------------
 SRID=4326;LINESTRING(110 40,2 3,-10 80,-7 9)
```

Amazon Redshift fornisce numerose funzioni SQL per eseguire query sui dati spaziali. Ad eccezione della funzione ST_IsValid, funzioni spaziali che accettano un oggetto GEOMETRY come argomento si aspettano che questo oggetto GEOMETRY sia una geometria valida. Se l'oggetto GEOMETRY o GEOGRAPHY non è valido, il comportamento della funzione spaziale non è definito. Per ulteriori informazioni sui nomi validi, consultare Validità geometrica.

Per i dettagli sulle funzioni SQL da usare per eseguire query sui dati spaziali, consultare Funzioni spaziali.

Per i dettagli sul caricamento di dati spaziali, consultare Caricamento di una colonna definita come tipo dati GEOMETRY o GEOGRAPHY.

Argomenti

- Tutorial: Utilizzo delle funzioni SQL spaziali con Amazon Redshift
- Caricamento di uno shapefile in Amazon Redshift
- Terminologia per i dati spaziali di Amazon Redshift
- Considerazioni sull'utilizzo dei dati spaziali con Amazon Redshift

Tutorial: Utilizzo delle funzioni SQL spaziali con Amazon Redshift

Questo tutorial mostra come utilizzare alcune delle funzioni SQL spaziali con Amazon Redshift.

A tale scopo, viene eseguita una query su due tabelle utilizzando le funzioni SQL spaziali. Il tutorial utilizza i dati provenienti da set di dati pubblici che correlano i dati sulla posizione degli alloggi in affitto con i codici postali a Berlino, Germania.

Argomenti

Prerequisiti

Per questo tutorial, sono necessarie le seguenti risorse:

- Un cluster e un database Amazon Redshift esistenti a cui è possibile accedere e aggiornare. Nel cluster esistente è possibile creare tabelle, caricare dati di esempio ed eseguire query SQL per dimostrare le funzioni spaziali. Il cluster deve avere almeno due nodi. Per informazioni su come creare un cluster, seguire la procedura riportata in Guida alle operazioni di base di Amazon Redshift.
- Per utilizzare l'editor di query Amazon Redshift, assicurarsi che il cluster si trovi in una regione AWS che supporta l'editor di query. Per ulteriori informazioni, consulta Esecuzione di query su un database con l'editor di query nella Guida alla gestione di Amazon Redshift.
- AWS credenziali per il tuo cluster Amazon Redshift che gli consentono di caricare i dati di test da Amazon S3. Per informazioni su come accedere ad altri AWS servizi come Amazon S3, consulta Autorizzazione di Amazon Redshift all'accesso ai servizi. AWS
- Il nome del ruolo AWS Identity and Access Management (IAM)`mySpatialDemoRole`, a cui è `AmazonS3ReadOnlyAccess` allegata la policy gestita per leggere i dati di Amazon S3. Per creare un ruolo con l'autorizzazione per caricare i dati da un bucket Amazon S3, consulta Autorizzazione delle operazioni COPY, UNLOAD e CREATE EXTERNAL SCHEMA mediante ruoli IAM nella Guida alla gestione di Amazon Redshift.
- Dopo aver creato il ruolo IAM `mySpatialDemoRole`, tale ruolo richiede un'associazione con il cluster Amazon Redshift. Per ulteriori informazioni su come creare tale associazione, consulta

Autorizzazione delle operazioni COPY, UNLOAD e CREATE EXTERNAL SCHEMA mediante ruoli IAM nella Guida alla gestione di Amazon Redshift.

Fase 1: Creazione di tabelle e caricamento dei dati di test

I dati di origine utilizzati in questo tutorial sono contenuti in file denominati `accommodations.csv` e `zipcodes.csv`.

Il file `accommodations.csv` è costituito da dati open source provenienti da insideairbnb.com. Il file `zipcodes.csv` fornisce codici postali che sono dati open source dell'istituto nazionale di statistica di Berlino-Brandeburgo in Germania (Amt für Statistik Berlino-Brandeburgo). Entrambe le origini dati sono fornite con una licenza Creative Commons. I dati sono limitati alla regione di Berlino, Germania. Questi file si trovano in un bucket pubblico Amazon S3 da utilizzare con questo tutorial.

Facoltativamente, è possibile scaricare i dati di origine dai seguenti link Amazon S3:

- Dati di origine per la tabella `accommodations`.
- Dati di origine per la tabella `zipcode`.

Utilizzare la procedura seguente per creare tabelle e caricare i dati di test.

Come creare tabelle e caricare dati di test

1. Aprire l'editor di query Amazon Redshift. Per ulteriori informazioni sull'utilizzo dell'editor di query, consulta Esecuzione di query su un database con l'editor della query nella Guida alla gestione di Amazon Redshift.
2. Eliminare le tabelle utilizzate da questo tutorial se sono già presenti nel database. Per ulteriori informazioni, consulta Fase 3: eliminazione delle risorse.
3. Creare la tabella `accommodations` per memorizzare la posizione geografica di ogni alloggio (longitudine e latitudine), il nome dell'inserzione e altri dati aziendali.

   Questo tutorial esplora gli affitti di camere a Berlino, in Germania. La colonna `shape` riporta i punti geografici della posizione degli alloggi. Le altre colonne contengono informazioni sull'affitto.

   Per creare la tabella `accommodations`, eseguire la seguente istruzione SQL nell'editor di query Amazon Redshift.

   ```
   CREATE TABLE public.accommodations (
   ```

```
  id INTEGER PRIMARY KEY,
  shape GEOMETRY,
  name VARCHAR(100),
  host_name VARCHAR(100),
  neighbourhood_group VARCHAR(100),
  neighbourhood VARCHAR(100),
  room_type VARCHAR(100),
  price SMALLINT,
  minimum_nights SMALLINT,
  number_of_reviews SMALLINT,
  last_review DATE,
  reviews_per_month NUMERIC(8,2),
  calculated_host_listings_count SMALLINT,
  availability_365 SMALLINT
);
```

4. Creare la tabella `zipcode` nell'editor di query per memorizzare i codici postali di Berlino.

   Un codice postale (o CAP) è definito come un poligono nella colonna `wkb_geometry`. Le altre colonne descrivono metadati spaziali aggiuntivi relativi al codice postale.

   Per creare la tabella `zipcode`, eseguire la seguente istruzione SQL nell'editor di query Amazon Redshift.

   ```
   CREATE TABLE public.zipcode (
     ogc_field INTEGER PRIMARY KEY NOT NULL,
     wkb_geometry GEOMETRY,
     gml_id VARCHAR(256),
     spatial_name VARCHAR(256),
     spatial_alias VARCHAR(256),
     spatial_type VARCHAR(256)
    );
   ```

5. Caricare le tabelle mediante i dati di esempio.

   I dati di esempio per questo tutorial sono forniti in un bucket Amazon S3 che consente l'accesso in lettura a tutti gli utenti autenticati. AWS Assicurarsi di fornire credenziali AWS valide che consentano di accedere ad Amazon S3.

   Per caricare i dati di test nelle tabelle, emettere i seguenti comandi COPY. Sostituire *`account-number`* con il numero di account AWS . Il segmento della stringa delle credenziali racchiuso tra virgolette singole non deve contenere spazi o interruzioni di riga.

```
COPY public.accommodations
FROM 's3://redshift-downloads/spatial-data/accommodations.csv'
DELIMITER ';'
IGNOREHEADER 1 REGION 'us-east-1'
CREDENTIALS 'aws_iam_role=arn:aws:iam::account-number:role/mySpatialDemoRole';
```

```
COPY public.zipcode
FROM 's3://redshift-downloads/spatial-data/zipcode.csv'
DELIMITER ';'
IGNOREHEADER 1 REGION 'us-east-1'
CREDENTIALS 'aws_iam_role=arn:aws:iam::account-number:role/mySpatialDemoRole';
```

6. Verificare che ogni tabella sia caricata correttamente emettendo i comandi seguenti.

```
select count(*) from accommodations;
```

```
select count(*) from zipcode;
```

I risultati seguenti mostrano il numero di righe di ogni tabella di dati di test.

Nome tabella	Righe
alloggi	22.248
zipcode	190

Fase 2: Query su dati spaziali

Dopo aver creato e caricato le tabelle, è possibile eseguire le query utilizzando le istruzioni SQL SELECT. Le query seguenti illustrano alcune delle informazioni che è possibile recuperare. È possibile scrivere molte altre query che utilizzano funzioni spaziali per soddisfare le proprie esigenze.

Come eseguire query su dati spaziali

1. Eseguire le query per ottenere il conteggio del numero totale di risultati memorizzati nella tabella `accommodations`, come illustrato di seguito. Il sistema di riferimento territoriale è World Geodetic System (WGS) 84, che ha l'identificatore univoco di riferimento territoriale 4326.

```
SELECT count(*) FROM public.accommodations WHERE ST_SRID(shape) = 4326;
```

```
 count
-------
 22248
```

2. Recuperare gli oggetti geometrici in formato WKT (well-known text) con alcuni attributi aggiuntivi. Inoltre, è possibile verificare se questi dati del codice postale sono memorizzati anche nel World Geodetic System (WGS) 84, che utilizza l'ID di riferimento spaziale (SRID) 4326. Perché possano essere utilizzati in maniera incrociata, i dati spaziali devono essere conservati nello stesso sistema di riferimento territoriale.

```
SELECT ogc_field, spatial_name, spatial_type, ST_SRID(wkb_geometry),
 ST_AsText(wkb_geometry)
FROM public.zipcode
ORDER BY spatial_name;
```

```
ogc_field  spatial_name  spatial_type  st_srid  st_astext
-------------------------------------------------------------
0           10115        Polygon        4326     POLYGON((...))
4           10117        Polygon        4326     POLYGON((...))
8           10119        Polygon        4326     POLYGON((...))
...
(190 rows returned)
```

3. Selezionare il poligono di Berlin Mitte (10117), un quartiere di Berlino, in formato GeoJSON, la sua dimensione e il numero di punti in questo poligono.

```
SELECT ogc_field, spatial_name, ST_AsGeoJSON(wkb_geometry),
 ST_Dimension(wkb_geometry), ST_NPoints(wkb_geometry)
FROM public.zipcode
WHERE spatial_name='10117';
```

```
ogc_field  spatial_name  spatial_type
 st_dimension  st_npoint
-------------------------------------------------------------------------------------------
```

```
4           10117         {"type":"Polygon", "coordinates":[[[...]]]}    2
    331
```

4. Emettere il seguente comando SQL per visualizzare quante strutture ricettive si trovano nel raggio di 500 metri dalla Porta di Brandeburgo.

```
SELECT count(*)
FROM public.accommodations
WHERE ST_DistanceSphere(shape, ST_GeomFromText('POINT(13.377704 52.516431)', 4326))
 < 500;
```

```
count
------
  29
```

5. Ottenere la posizione approssimativa della Porta di Brandeburgo dai dati archiviati nelle strutture elencate come nelle vicinanze eseguendo la seguente query.

   Questa query richiede una sottoselezione. Porta a un conteggio diverso perché la posizione richiesta non è la stessa della query precedente perché è più vicina agli alloggi.

```
WITH poi(loc) as (
  SELECT st_astext(shape) FROM accommodations WHERE name LIKE '%brandenburg gate%'
)
SELECT count(*)
FROM accommodations a, poi p
WHERE ST_DistanceSphere(a.shape, ST_GeomFromText(p.loc, 4326)) < 500;
```

```
count
------
  60
```

6. Eseguire la seguente query per mostrare i dettagli di tutti gli alloggi intorno alla Porta di Brandeburgo, ordinati per prezzo in ordine decrescente.

```
SELECT name, price, ST_AsText(shape)
FROM public.accommodations
WHERE ST_DistanceSphere(shape, ST_GeomFromText('POINT(13.377704 52.516431)', 4326))
 < 500
```

```
ORDER BY price DESC;
```

```
name                                                          price  st_astext
------------------------------------------------------------------------------------------------------------
DUPLEX APARTMENT/PENTHOUSE in 5* LOCATION! 7583               300
 POINT(13.3826510209548 52.5159819722552)
DUPLEX-PENTHOUSE IN FIRST LOCATION! 7582                      300
 POINT(13.3799997083855 52.5135918444834)
...
(29 rows returned)
```

7. Eseguire la seguente query per recuperare la sistemazione più costosa con il suo codice postale.

```
SELECT
  a.price, a.name, ST_AsText(a.shape),
  z.spatial_name, ST_AsText(z.wkb_geometry)
FROM accommodations a, zipcode z
WHERE price = 9000 AND ST_Within(a.shape, z.wkb_geometry);
```

```
price   name                                    st_astext
     spatial_name      st_astext
------------------------------------------------------------------------------------------------------------
9000    Ueber den Dächern Berlins Zentrum    POINT(13.334436985013
 52.4979779501538)    10777             POLYGON((13.3318284987227
 52.4956021172799,...
```

8. Calcolare il prezzo massimo, minimo o medio degli alloggi utilizzando una sottoquery.

   La query seguente elenca il prezzo medio degli alloggi in base al codice postale.

```
SELECT
  a.price, a.name, ST_AsText(a.shape),
  z.spatial_name, ST_AsText(z.wkb_geometry)
FROM accommodations a, zipcode z
WHERE
  ST_Within(a.shape, z.wkb_geometry) AND
  price = (SELECT median(price) FROM accommodations)
ORDER BY a.price;
```

```
price name                              st_astext
 spatial_name   st_astext
-------------------------------------------------------------------------------------------

45    "Cozy room Berlin-Mitte"        POINT(13.3864349535358 52.5292016386514)
 10115          POLYGON((13.3658598465795 52.535659581048,...
...
(723 rows returned)
```

9. Eseguire la seguente query per recuperare il numero di alloggi elencati a Berlino. Per trovare gli hot spot, questi sono raggruppati per codice postale e ordinati in base alla quantità di fornitura.

```
SELECT z.spatial_name as zip, count(*) as numAccommodations
FROM public.accommodations a, public.zipcode z
WHERE ST_Within(a.shape, z.wkb_geometry)
GROUP BY zip
ORDER BY numAccommodations DESC;
```

```
zip   numaccommodations
----------------------------
10245 872
10247 832
10437 733
10115 664
...
(187 rows returned)
```

Fase 3: eliminazione delle risorse

Il cluster genera dei costi fino a che è in esecuzione. Una volta completato questo tutorial, sarà possibile eliminare il cluster di esempio.

Se desideri conservare il cluster, ma recuperare l'archiviazione utilizzata dalle tabelle di dati di test, emettere i comandi seguenti per eliminare le tabelle.

```
drop table public.accommodations cascade;
```

```
drop table public.zipcode cascade;
```

Caricamento di uno shapefile in Amazon Redshift

È possibile utilizzare il comando COPY per importare gli shapefile Esri archiviati in Amazon S3 nelle tabelle Amazon Redshift. Uno shapefile memorizza le informazioni sulla posizione geometrica e sugli attributi delle caratteristiche geografiche in un formato vettoriale. Il formato shapefile può descrivere spazialmente oggetti spaziali quali punti, linee e poligoni. Per ulteriori informazioni su uno shapefile, consultare Shapefile in Wikipedia.

Il comando COPY supporta il parametro del formato dati `SHAPEFILE`. Per impostazione predefinita, la prima colonna dello shapefile è una colonna `GEOMETRY` o `IDENTITY`. Tutte le colonne successive seguono l'ordine specificato nello shapefile. Tuttavia, la tabella di destinazione non deve avere questo layout esatto perché è possibile utilizzare la mappatura delle colonne COPY per definire l'ordine. Per informazioni sul supporto dello shapefile del comando COPY, consultare SHAPEFILE.

In alcuni casi, la dimensione della geometria risultante potrebbe essere maggiore del massimo per la memorizzazione di una geometria in Amazon Redshift. In tal caso, è possibile utilizzare l'opzione COPY `SIMPLIFY` o `SIMPLIFY AUTO` per semplificare le geometrie durante l'importazione come riportato di seguito:

- Specificare `SIMPLIFY` *`tolerance`* per semplificare tutte le geometrie durante l'importazione dati utilizzando l'algoritmo Ramer-Douglas-Peucker e la tolleranza specificata.
- Specificare `SIMPLIFY AUTO` senza tolleranza per semplificare solo le geometrie che sono più grandi della dimensione massima utilizzando l'algoritmo Ramer-Douglas-Peucker. Questo approccio calcola la tolleranza minima sufficientemente grande da memorizzare l'oggetto entro il limite di dimensione massima.
- Specificare `SIMPLIFY AUTO` *`max_tolerance`* per semplificare solo le geometrie che sono più grandi della dimensione massima utilizzando l'algoritmo Ramer-Douglas-Peucker e la tolleranza calcolata automaticamente. Questo approccio assicura che la tolleranza non superi la tolleranza massima.

Per informazioni sulle dimensioni massime di un valore di dati `GEOMETRY`, consultare Considerazioni sull'utilizzo dei dati spaziali con Amazon Redshift.

In alcuni casi, la tolleranza è sufficientemente bassa che il record non può ridursi al di sotto della dimensione massima di un valore di dati GEOMETRY. In questi casi, è possibile utilizzare l'opzione MAXERROR del comando COPY per ignorare tutti o fino a un certo numero di errori di inserimento.

Il comando COPY supporta anche il caricamento degli shapefile GZIP. Per eseguire questa operazione, specificare il parametro COPY GZIP. Con questa opzione, tutti i componenti shapefile devono essere compressi in modo indipendente e condividere lo stesso suffisso di compressione.

Se esiste un file di descrizione della proiezione (.prj) con lo shapefile, Redshift lo utilizza per determinare l'ID del sistema di riferimento spaziale (SRID). Se lo SRID è valido, alla geometria risultante viene assegnato questo SRID. Se il valore SRID associato alla geometria di input non esiste, la geometria risultante avrà il valore dello SRID uguale a zero. È possibile disabilitare il rilevamento automatico dell'ID del sistema di riferimento spaziale a livello di sessione utilizzando SET read_srid_on_shapefile_ingestion su OFF.

Eseguire query sul sistema SYS_SPATIAL_SIMPLIFY o SVL_SPATIAL_SIMPLIFY per visualizzare quali record sono stati semplificati, insieme alla tolleranza calcolata. Quando si specifica SIMPLIFY *tolerance*, questa vista contiene un record per ogni operazione COPY. In caso contrario, contiene un record per ogni geometria semplificata. Per ulteriori informazioni, consulta SYS_SPATIAL_SIMPLIFY o SVL_SPATIAL_SIMPLIFY.

Per esempi di caricamento di uno shapefile, consultare Caricamento di uno shapefile in Amazon Redshift.

Terminologia per i dati spaziali di Amazon Redshift

I seguenti termini vengono utilizzati per descrivere alcune funzioni spaziali di Amazon Redshift.

Riquadro di delimitazione

Un riquadro di delimitazione di una geometria o di un'area geografica è definito come il prodotto incrociato (attraverso le dimensioni) delle estensioni delle coordinate di tutti i punti della geometria o dell'area geografica. Per le geometrie bidimensionali, il rettangolo di delimitazione è un rettangolo che include completamente tutti i punti della geometria. Ad esempio, un rettangolo di selezione del poligono POLYGON((0 0,1 0,0 2,0 0)) è il rettangolo definito dai punti (0, 0) e (1, 2) come gli angoli in basso a sinistra e in alto a destra. Amazon Redshift precalcola e memorizza un riquadro di delimitazione all'interno di una geometria per velocizzare i predicati geometrici e i join spaziali. Ad esempio, se i riquadri di delimitazione di due geometrie non si intersecano, queste due geometrie

non possono intersecarsi e non possono trovarsi nel set di risultati di un join spaziale utilizzando il predicato ST_Intersects.

È possibile utilizzare le funzioni spaziali per aggiungere (AddBBox), eliminare (DropBBox) e determinare il supporto (SupportsBBox) per un riquadro di delimitazione. Amazon Redshift supporta il precalcolo dei riquadri di delimitazione per tutti i sottotipi di geometria.

L'esempio seguente mostra come aggiornare le geometrie esistenti in una tabella per memorizzarle con un riquadro di delimitazione. Se il cluster è alla versione 1.0.26809 o successiva, tutte le nuove geometrie vengono create con un riquadro di delimitazione precalcolato per impostazione predefinita.

```
UPDATE my_table SET geom = AddBBox(geom) WHERE SupportsBBox(geom) = false;
```

Dopo aver aggiornato le geometrie esistenti, si consiglia di eseguire il comando VACUUM sulla tabella aggiornata. Per ulteriori informazioni, consulta VACUUM.

Per impostare se le geometrie sono codificate con un riquadro di delimitazione durante una sessione, consultare default_geometry_encoding.

Validità geometrica

Gli algoritmi geometrici utilizzati da Amazon Redshift presuppongono che la geometria di input sia una geometria valida. Se un input a un algoritmo non è valido, il risultato non è definito. La sezione seguente descrive le definizioni di validità geometrica utilizzate da Amazon Redshift per ciascun sottotipo di geometria.

Point (Punto)

Un punto è considerato valido se una delle seguenti condizioni è vera:

- Il punto è il punto vuoto.
- Tutte le coordinate dei punti sono numeri a virgola mobile finiti.

Un punto può essere il punto vuoto.

Linestring

Una linestring è considerata valida se una o più delle seguenti condizioni è vera:

- La linestring è vuota, ovvero non contiene punti.
- Tutti i punti in una linestring non vuoto hanno coordinate che sono numeri a virgola mobile finiti.

- La linestring, se non vuota, deve essere unidimensionale, ovvero non può degenerare su un punto.

Una linestring non può contenere punti vuoti.

Una linestring può avere punti consecutivi duplicati.

Una linestring può avere auto-intersezioni.

Polygon

Un poligono è considerato valido se una o più delle seguenti condizioni è vera:

- La linestring è vuota, ovvero non contiene anelli.
- Se non è vuoto, un poligono è valido se si verificano tutte le condizioni seguenti:
  - Tutti gli anelli del poligono sono validi. Un anello è considerato valido se una o più delle seguenti condizioni è vera:
    - Tutti i punti dell'anello non vuoto hanno coordinate che sono numeri a virgola mobile finiti.
    - L'anello è chiuso, cioè il suo primo punto e il suo ultimo punto coincidono.
    - L'anello non ha auto-intersezioni.
    - L'anello è bidimensionale.
  - Gli anelli del poligono hanno orientamenti coerenti. In altre parole, se si attraversa un qualsiasi anello, l'interno del poligono è alla destra o alla sinistra dell'utente. Ciò significa che se l'anello esterno di un poligono è orientato in senso orario o antiorario, tutti gli anelli interni del poligono devono avere lo stesso orientamento in senso antiorario o orario.
  - Tutti gli anelli interni devono trovarsi all'interno dell'anello esterno del poligono.
  - Gli anelli interni non possono essere nidificati, ovvero un anello interno non può trovarsi all'interno di un altro anello interno.
  - Gli anelli interni ed esterni possono intersecarsi solo in un numero limitato di punti.
  - L'interno del poligono deve essere collegato semplicemente.

Un poligono non può contenere punti vuoti.

Multipoint

Un multipunto è considerato valido se una o più delle seguenti condizioni è vera:

- Il multipunto è vuoto, ovvero non contiene punti.
- Un multipunto non è vuoto e tutti i punti sono validi in base alla definizione di validità del punto.

Un multipunto può contenere uno o più punti vuoti.

Un multipunto può avere punti duplicati.

Multiinestring

Una funzione multilinestring è considerata valida se una o più delle seguenti condizioni è vera:

- La multilinestring è vuota, ovvero non contiene punti.
- Tutte le linee in una multilinestring non vuota sono valide in base alla definizione di validità della linea.

Una multilinestring non vuota costituita solo da linee vuote è considerata valida.

Una linestring vuota in una multilinestring non influisce sulla sua validità.

Una multilinestring può avere linestring con punti consecutivi duplicati.

Una multilinestring può avere auto-intersezioni.

Una multilinestring non può contenere punti vuoti.

Multipolygon

Un multipoligono è considerato valido se una o più delle seguenti condizioni è vera:

- Il multipoligono non contiene poligoni (è vuoto).
- Il multipoligono non è vuoto e quanto riportato di seguito è vero:
  - Tutti i poligoni nel multipoligono sono validi.
  - Due poligoni nel multipoligono non possono intersecarsi in un numero infinito di punti. In particolare, ciò implica che l'interno di due poligoni non può intersecarsi e che possono toccare solo in un numero finito di punti.

Un poligono vuoto in un multipoligono non invalida un multipoligono.

Un multipoligono non può contenere punti vuoti.

Raccolta geometrica

Una raccolta geometrica è considerata valida se una o più delle seguenti condizioni è vera:

- La raccolta geometrica è vuota, ovvero non contiene geometrie.
- Tutte le geometrie in una raccolta geometrica non vuota sono valide.

Questa definizione si applica ancora, anche se in modo ricorsivo, per le raccolte geometriche nidificate.

Una raccolta geometrica può contenere punti vuoti e multipunto con punti vuoti.

Semplicità geometrica

Gli algoritmi geometrici utilizzati da Amazon Redshift presuppongono che la geometria di input sia una geometria valida. Se un input a un algoritmo non è valido, il controllo della semplicità non è definito. La sezione seguente descrive le definizioni di semplicità geometrica utilizzate da Amazon Redshift per ciascun sottotipo di geometria.

Point (Punto)

Un punto valido è considerato semplice se una o più delle seguenti condizioni è vera:

- Un punto valido è sempre considerato semplice.
- Un punto vuoto è considerato semplice.

Linestring

Una linestring è considerata semplice se una o più delle seguenti condizioni è vera:

- La linestring è vuota.
- La linestring non è vuota e si verificano tutte le condizioni seguenti:
  - Non ha punti consecutivi duplicati.
  - Non ha auto-intersezioni, tranne possibilmente per il suo primo punto e ultimo punto, che possono coincidere. In altre parole, la linestring non può avere auto-intersezioni tranne nei punti limite.

Polygon

Un poligono valido è considerato semplice se non contiene punti consecutivi duplicati.

Multipoint

Un multipunto valido è considerato semplice se una o più delle seguenti condizioni è vera:

- Il multipunto è vuoto, ovvero non contiene punti.
- Due punti non vuoti del multipunto non coincidono.

Multilinestring

Una multilinestring valida è considerata semplice se una o più delle seguenti condizioni è vera:

- La multilinestring è vuota.
- La multilinestring non è vuota e si verificano tutte le condizioni seguenti:
  - Tutte le relative linestring sono semplici.
  - Qualsiasi due linestring della multilinestring non si intersecano, tranne nei punti che sono punti limite delle due linee.

Una multilinestring non vuota costituita solo da linestring vuote è considerata vuota.

Una linestring vuota in una multilinestring non influisce sulla sua semplicità.

Una linestring chiusa in una multilinestring non può intersecarsi con qualsiasi altra linestring nella multilinestring.

Una multilinestring non può avere linestring con punti consecutivi duplicati.

Multipolygon

Un multipoligono valido è considerato semplice se non contiene punti consecutivi duplicati.

Raccolta geometrica

Una raccolta geometrica valida è considerata semplice se una o più delle seguenti condizioni è vera:

- La raccolta geometrica è vuota, ovvero non contiene geometrie.
- Tutte le geometrie in una raccolta geometrica non vuota sono semplici.

Questa definizione si applica ancora, anche se in modo ricorsivo, per le raccolte geometriche nidificate.

H3

H3 è un sistema a griglia di indicizzazione geospaziale gerarchico che offre un modo per indicizzare le coordinate spaziali fino a una risoluzione di metro quadrato. I dati indicizzati possono essere uniti tra diversi set di dati e aggregati a diversi livelli di precisione. H3 consente una serie di algoritmi e ottimizzazioni basati sulla griglia, tra cui i cui vicini più prossimi, il percorso più breve, l'ottimizzazione dei gradienti e altro ancora. Gli indici H3 fanno riferimento a celle che possono essere esagoni o pentagoni. Lo spazio è suddiviso gerarchicamente in base a una risoluzione. H3 supporta 16 risoluzioni da 0 a 15, entrambi compresi. di cui `0` è la risoluzione più grossolana e `15` la più granulare.

Amazon Redshift fornisce le seguenti funzioni spaziali H3:

- H3_ FromLongLat
- H3_ FromPoint
- H3_Polyfill

Considerazioni sull'utilizzo dei dati spaziali con Amazon Redshift

Le seguenti sono considerazioni relative all'utilizzo dei dati spaziali con Amazon Redshift:

- La dimensione massima di un oggetto GEOMETRY o GEOGRAPHY è di 1.048.447 byte.
- Amazon Redshift Spectrum non supporta nativamente i dati spaziali. Per questo motivo non è possibile creare o modificare una tabella esterna con una colonna di tipo GEOMETRY o GEOGRAPHY.
- I tipi di dati per le funzioni definite dall'utente (UDF) in Python non supportano il tipo di dati GEOMETRY o GEOGRAPHY.
- Non è possibile utilizzare una colonna GEOMETRY o GEOGRAPHY come chiave di ordinamento o chiave di distribuzione di una tabella Amazon Redshift.
- Non è possibile utilizzare colonne di tipo GEOMETRY o GEOGRAPHY nelle clausole SQL ORDER BY, GROUP BY o DISTINCT.
- Non è possibile utilizzare colonne di tipo GEOMETRY o GEOGRAPHY in molte funzioni SQL.
- In caso di colonne GEOMETRY o GEOGRAPHY non è possibile eseguire un'operazione di UNLOAD in qualsiasi formato. È possibile eseguire il comando UNLOAD sulle colonne GEOMETRY o GEOGRAPHY nei file di testo o CSV (valori separati da virgole). In questo modo scrive i dati GEOMETRY o GEOGRAPHY in formato esadecimale EWKB. Se la dimensione dei dati in formato EWKB è superiore a 4 MB, l'utente riceve l'avviso del fatto che i dati non possono essere successivamente caricati in una tabella.
- La codifica di compressione supportata dai dati di tipo GEOMETRY o GEOGRAPHY è RAW.
- Quando si utilizzano driver JDBC o ODBC, utilizzare le mappature dei tipi personalizzate. In questo caso, l'applicazione client deve disporre delle informazioni su quali parametri di un oggetto ResultSet sono oggetti di tipo GEOMETRY o GEOGRAPHY. L'operazione ResultSetMetadata restituisce un valore di tipo VARCHAR.
- Per copiare la data geografica da un SHAPEFILE, prima ingerire in un colonna GEOMETRY, quindi lanciare gli oggetti su oggetti GEOGRAPHY. .

Le seguenti funzioni non spaziali possono accettare un input di tipo GEOMETRY o GEOGRAPHY, o colonne di tipo GEOMETRY o GEOGRAPHY:

- La funzione di aggregazione COUNT
- Le espressioni condizionali COALESCE e NVL
- Espressioni CASE
- La codifica di default per `GEOMETRY` e `GEOGRAPHY` è RAW. Per ulteriori informazioni, consulta Codifiche di compressione.

Esecuzione di query su dati con query federate in Amazon Redshift

L'uso delle query federate in Amazon Redshift consente di eseguire query e analizzare i dati in database operativi, data warehouse e data lake. Con la funzionalità di query federata, è possibile integrare query da Amazon Redshift in tempo reale in database esterni con query in tutti gli ambienti Amazon Redshift ed Amazon S3. Le query federate possono funzionare con database esterni in Amazon RDS for PostgreSQL, Amazon Aurora Edizione compatibile con PostgreSQL, Amazon RDS for MySQL (anteprima) e Amazon Aurora Edizione compatibile con MySQL (anteprima).

Le query federate ti consentono di incorporare dati in tempo reale come parte delle applicazioni di business intelligence (BI) e reporting. Ad esempio, per semplificare l'importazione di dati in Amazon Redshift, è possibile utilizzare query federate per effettuare le seguenti operazioni:

- Interrogare direttamente i database operativi.
- Applicare rapidamente le trasformazioni.
- Caricare i dati nelle tabelle di destinazione senza la necessità di pipeline complesse di estrazione, trasformazione, caricamento (ETL).

Per ridurre lo spostamento dei dati sulla rete e migliorare le prestazioni, Amazon Redshift distribuisce parte del calcolo per le query federate direttamente nei database operativi remoti. Se necessario, per supportare l'esecuzione di queste query, Amazon Redshift utilizza anche la sua capacità di elaborazione parallela.

Quando si eseguono query federate, Amazon Redshift effettua innanzitutto una connessione client all'istanza DB del cluster RDS o Aurora DB dal nodo leader per recuperare i metadati della tabella. Da un nodo di calcolo, Amazon Redshift emette query secondarie con un predicato spinto verso il basso e recupera le righe dei risultati. Quindi Amazon Redshift distribuisce le righe dei risultati tra i nodi di calcolo per ulteriori elaborazioni.

I dettagli sulle query inviate al database Amazon Aurora PostgreSQL o al database Amazon RDS for PostgreSQL vengono registrati nella vista di sistema SVL_FEDERATED_QUERY.

Argomenti

- Nozioni di base sull'utilizzo di query federate su PostgreSQL
- Iniziare a utilizzare le query federate su PostgreSQL con AWS CloudFormation

- Nozioni di base sull'utilizzo di query federate su MySQL (anteprima)
- Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate
- Esempio di utilizzo di una query federata
- Differenze dei tipi di dati tra Amazon Redshift e database PostgreSQL e MySQL supportati
- Considerazioni quando si accede ai dati federati con Amazon Redshift

Nozioni di base sull'utilizzo di query federate su PostgreSQL

Per creare una query federata, segui questo approccio generale:

1. Configurare la connettività dal cluster Amazon Redshift all'istanza database Amazon RDS o Aurora PostgreSQL.

   A tale scopo, assicurarsi che l'istanza database di RDS PostgreSQL o Aurora PostgreSQL possa accettare connessioni dal cluster Amazon Redshift. È consigliabile che il cluster Amazon Redshift e l'istanza Amazon RDS o Aurora PostgreSQL siano nello stesso virtual private cloud (VPC) e gruppo di sottoreti. In questo modo, è possibile aggiungere il gruppo di sicurezza per il cluster Amazon Redshift alle regole in ingresso del gruppo di sicurezza per l'istanza database RDS o Aurora PostgreSQL.

   È possibile inoltre impostare il peering VPC o altre reti che consentono ad Amazon Redshift di effettuare connessioni all'istanza RDS o Aurora PostgreSQL. Per ulteriori informazioni sulla rete VPC, consulta quanto segue.

   - Che cos'è il peering di VPC? nella Guida Amazon per il peering VPC
   - Uso di un'istanza database in un VPC nella Guida per l'utente di Amazon RDS

   > **Note**
   >
   > In alcuni dei seguenti casi devi abilitare l'instradamento avanzato del VPC: ad esempio, se il cluster Amazon Redshift si trova in un VPC diverso da quello dell'istanza RDS o Aurora PostgreSQL o se si trovano nello stesso VPC e l'instradamento lo richiede. In caso contrario, è possibile che vengano visualizzati errori di timeout quando si esegue una query federata.

2. Imposta i segreti AWS Secrets Manager per i tuoi database RDS PostgreSQL e Aurora PostgreSQL. Quindi fai riferimento ai segreti nelle AWS Identity and Access Management politiche

e nei ruoli di accesso (IAM). Per ulteriori informazioni, consulta Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate.

> **Note**
>
> Se il cluster utilizza il routing VPC avanzato, potrebbe essere necessario configurare un endpoint VPC di interfaccia per AWS Secrets Manager. Ciò è necessario quando il VPC e la sottorete del cluster Amazon Redshift non hanno accesso all'endpoint pubblico. AWS Secrets Manager Quando utilizzi un endpoint con interfaccia VPC, la comunicazione tra il cluster Amazon Redshift e il tuo VPC viene AWS Secrets Manager instradata privatamente dal tuo VPC all'interfaccia endpoint. Per ulteriori informazioni, consultare Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

3. Applicare il ruolo IAM creato in precedenza al cluster Amazon Redshift. Per ulteriori informazioni, consultare Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate.
4. Connettersi ai database RDS PostgreSQL e Aurora PostgreSQL con uno schema esterno. Per ulteriori informazioni, consultare CREATE EXTERNAL SCHEMA. Per esempi su come utilizzare la query federata, consultare Esempio di utilizzo di una query federata.
5. Eseguire le query SQL che fanno riferimento allo schema esterno che fa riferimento ai database RDS PostgreSQL e Aurora PostgreSQL.

Iniziare a utilizzare le query federate su PostgreSQL con AWS CloudFormation

È possibile utilizzare query federate per eseguire query su database operativi. In questa guida introduttiva, puoi automatizzare la configurazione utilizzando uno AWS CloudFormation stack di esempio per abilitare una query federata da un cluster Amazon Redshift a un database serverless Aurora PostgreSQL. Pertanto, è possibile eseguire rapidamente l'esecuzione senza dover eseguire istruzioni SQL per effettuare il provisioning delle risorse.

La pila crea uno schema esterno che fa riferimento all'istanza Aurora PostgreSQL, che include tabelle con dati di esempio. È possibile eseguire query sulle tabelle nello schema esterno dal cluster Redshift.

Se invece desideri iniziare con le query federate eseguendo istruzioni SQL per configurare uno schema esterno, senza utilizzarlo, consulta. CloudFormation Nozioni di base sull'utilizzo di query federate su PostgreSQL

Prima di eseguire lo CloudFormation stack per le query federate, assicurati di disporre di un database serverless Edition compatibile con Amazon Aurora PostgreSQL con l'API Data attivata. È possibile attivare l'API dati nelle proprietà del database. Se non riesci a trovare l'impostazione, controlla di aver eseguito un'istanza serverless di Aurora PostgreSQL. Assicurati inoltre di disporre di un cluster Amazon Redshift che utilizza nodi RA3. È consigliabile che il cluster Redshift e l'istanza Aurora PostgreSQL serverless siano nello stesso cloud privato virtuale (VPC) e gruppo di sottoreti. In questo modo, è possibile aggiungere il gruppo di sicurezza per il cluster Amazon Redshift alle regole in ingresso del gruppo di sicurezza per l'istanza database RDS o Aurora PostgreSQL.

Per ulteriori informazioni su come iniziare a configurare un cluster Amazon Redshift, consulta Amazon Redshift provisioned clusters. Per ulteriori informazioni sulla configurazione delle risorse con CloudFormation, consulta What is? AWS CloudFormation . Per ulteriori informazioni sulla configurazione di un database cluster Aurora DB, vedere Creazione di un cluster Aurora DB Serverless v1 DB cluster.

Avvio di uno CloudFormation stack per le query federate di Redshift

Utilizza la seguente procedura per avviare lo CloudFormation stack per Amazon Redshift e abilitare le query federate. Prima di farlo, assicurati di aver configurato il cluster Amazon Redshift e l'istanza Aurora PostgreSQL serverless.

Per avviare lo stack per le query federate CloudFormation

1. Fai clic su Avvia lo stack CFN qui per avviare il servizio in. CloudFormation AWS Management Console

   Se ti viene richiesto, effettua l'accesso.

   Viene avviato il processo di creazione dello stack, facendo riferimento a un file CloudFormation modello archiviato in Amazon S3. Un CloudFormation modello è un file di testo in formato JSON che dichiara AWS le risorse che compongono uno stack.
2. Scegliere Successivo per inserire i dettagli della pila.
3. In Parametri, per il cluster, immettere quanto segue:

   - Il nome del cluster Amazon Redshift, ad esempio **`ra3-consumer-cluster`**
   - Un nome di database specifico, ad esempio **`dev`**
   - Il nome utente di database, ad esempio **`consumeruser`**

Immettete anche i parametri per il database del cluster Aurora DB, inclusi l'utente, il nome del database, la porta e l'endpoint. Si consiglia di utilizzare un cluster di prova e testare il database serverless, poiché la pila crea diversi oggetti di database.

Seleziona Successivo.

Vengono visualizzate le opzioni della pila.

4. Scegliere Successivo per accettare le impostazioni predefinite.
5. In Capacità, scegli Riconosco che AWS CloudFormation potrebbe creare risorse IAM.
6. Seleziona Crea stack.

Scegli Create stack. CloudFormation effettua il provisioning delle risorse del modello, operazione che richiede circa 10 minuti, e crea uno schema esterno.

Se si verifica un errore durante la creazione della pila, procedere come segue:

- Visualizza la scheda CloudFormation Eventi per informazioni che possono aiutarti a risolvere l'errore.
- Assicurarsi di aver immesso il nome, il nome del database e il nome utente del database corretti per il cluster Redshift. Controllare anche i parametri per l'istanza Aurora PostgreSQL.
- Assicurarsi che il cluster disponga di nodi RA3.
- Assicurati che il database e il cluster Redshift si trovino nella stessa sottorete e gruppo di sicurezza.

Interrogazione di dati dallo schema esterno

Per utilizzare la procedura seguente, assicurarsi di disporre delle autorizzazioni necessarie per l'esecuzione di query sul cluster e sul database descritto.

Per eseguire query su un database esterno con query federata

1. Connettersi al database Redshift immesso al momento della creazione della pila, utilizzando uno strumento client come l'editor di query Redshift.
2. Interrogazione dello schema esterno creato dalla pila.

```
select * from svv_external_schemas;
```

La visualizzazione SVV_EXTERNAL_SCHEMAS restituisce informazioni sugli schemi esterni disponibili. In questo caso, viene restituito lo schema esterno creato dalla pila, `myfederated_schema`. È possibile che vengano restituiti anche altri schemi esterni, se avete delle impostazioni attive. La vista restituisce anche il database associato allo schema. Il database è il database cluster Aurora DB che hai inserito quando hai creato lo stack. Lo stack aggiunge una tabella al database del cluster Aurora DB, `category` chiamata, e un'altra tabella chiamata. `sales`

3. Esegui query SQL sulle tabelle nello schema esterno che fa riferimento al database Aurora PostgreSQL. Il seguente esempio mostra una query.

```
SELECT count(*) FROM myfederated_schema.category;
```

La tabella `category` restituisce diversi registri. È inoltre possibile restituire i registri dalla tabella `sales`.

```
SELECT count(*) FROM myfederated_schema.sales;
```

Per ulteriori esempi, consulta Esempio di utilizzo di una query federata.

Nozioni di base sull'utilizzo di query federate su MySQL (anteprima)

Per creare una query federata su database MySQL, seguire questo approccio generale:

1. Configurare la connettività dal cluster Amazon Redshift all'istanza database Amazon RDS o Aurora PostgreSQL.

   A tale scopo, assicurarsi che l'istanza database di RDS PostgreSQL o Aurora MySQL possa accettare connessioni dal cluster Amazon Redshift. È consigliabile che il cluster Amazon Redshift e l'istanza Amazon RDS o Aurora MySQL siano nello stesso Virtual Private Cloud (VPC) e gruppo di sottoreti. In questo modo, è possibile aggiungere il gruppo di sicurezza per il cluster Amazon Redshift alle regole in ingresso del gruppo di sicurezza per l'istanza database RDS o Aurora MySQL.

È inoltre possibile configurare il peering VPC o altre reti che consentono ad Amazon Redshift di effettuare connessioni all'istanza RDS o Aurora MySQL. Per ulteriori informazioni sulla rete VPC, consulta quanto segue.

- Che cos'è il peering di VPC? nella Guida Amazon per il peering VPC
- Uso di un'istanza database in un VPC nella Guida per l'utente di Amazon RDS

> **Note**
>
> Se il cluster Amazon Redshift si trova in un VPC diverso da quello dell'istanza RDS o Aurora PostgreSQL, abilitare il routing VPC avanzato. In caso contrario, è possibile che vengano visualizzati errori di timeout quando si esegue una query federata.

2. Configura segreti AWS Secrets Manager per i tuoi database RDS MySQL e Aurora MySQL. Quindi fai riferimento ai segreti nelle politiche e nei ruoli di accesso AWS Identity and Access Management (IAM). Per ulteriori informazioni, consulta Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate.

> **Note**
>
> Se il cluster utilizza il routing VPC avanzato, potrebbe essere necessario configurare un endpoint VPC di interfaccia per AWS Secrets Manager. Ciò è necessario quando il VPC e la sottorete del cluster Amazon Redshift non hanno accesso all'endpoint pubblico. AWS Secrets Manager Quando si utilizza un endpoint dell'interfaccia VPC, la comunicazione tra il cluster Amazon Redshift nel VPC e AWS Secrets Manager viene instradata privatamente dal VPC all'interfaccia dell'endpoint. Per ulteriori informazioni, consultare Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

3. Applicare il ruolo IAM creato in precedenza al cluster Amazon Redshift. Per ulteriori informazioni, consultare Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate.
4. Connettiti ai database PostgreSQL RDS e Aurora MySQL con uno schema esterno. Per ulteriori informazioni, consultare CREATE EXTERNAL SCHEMA. Per esempi su come utilizzare le query federate, consultare Esempio di utilizzo di una query federata con MySQL.
5. Eseguire le query SQL che fanno riferimento allo schema esterno che fa riferimento ai database RDS MySQL e Aurora MySQL.

Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate

La procedura seguente mostra come creare un segreto e un ruolo IAM da utilizzare con le query federate.

Prerequisiti

Assicurati di disporre dei seguenti prerequisiti per creare un segreto e un ruolo IAM da utilizzare con la query federata:

- Una istanza database RDS PostgreSQL, Aurora PostgreSQL, RDS MySQL o Aurora MySQL con autenticazione con nome utente e password.
- Un cluster Amazon Redshift con una versione di manutenzione cluster che supporta query federate.

Per creare un segreto (nome utente e password) con AWS Secrets Manager

1. Accedi alla console Secrets Manager con l'account che possiede l'istanza del cluster RDS o Aurora DB.
2. Scegli Archivia un nuovo segreto.
3. Scegliere il riquadro Credenziali per il database RDS . Per Nome utente e Password, immettere i valori per l'istanza. Confermare o scegliere un valore per Chiave di crittografia. Quindi scegliere il database RDS a cui avrà accesso il segreto.

   > **Note**
   >
   > Consigliamo di utilizzare la chiave di crittografia predefinita (`DefaultEncryptionKey`). Se usi una chiave di crittografia personalizzata, è necessario aggiungere il ruolo IAM utilizzato per accedere al segreto come utente chiave.

4. Immettere un nome per il segreto, continuare con la procedura di creazione con le scelte predefinite e poi scegliere Archivia.
5. Visualizzare il segreto e prendere nota del valore ARN del segreto creato per identificare il segreto.

Per creare una policy di sicurezza utilizzando il segreto

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Creare una policy con un codice JSON simile al seguente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessSecret",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-west-2:123456789012:secret:my-
rds-secret-VNenFy"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword",
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*"
        }
    ]
}
```

   Per recuperare il segreto sono necessarie operazioni di elenco e lettura. Si consiglia di limitare la risorsa al segreto specifico creato. A tale scopo, utilizzare l'Amazon Resource Name (ARN) del segreto per limitare la risorsa. È inoltre possibile specificare le autorizzazioni e le risorse utilizzando l'editor visivo nella console IAM.
3. Assegnare un nome alla policy e completare la creazione.
4. Passare a IAM roles (Ruoli IAM).
5. Creare un ruolo IAM per Redshift - Customizable (Redshift - Personalizzabile).

6. Collegare la policy IAM appena creata a un ruolo IAM esistente oppure creare un nuovo ruolo IAM a cui collegarla.
7. Nella scheda Trust relationships (Relazioni attendibili) del ruolo IAM confermare che il ruolo contenga l'entità di attendibilità `redshift.amazonaws.com`.
8. Prendere nota del Role ARN (Ruolo ARN) creato. Questo ARN ha accesso al segreto.

Come collegare il ruolo IAM al cluster Amazon Redshift

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster). Sono elencati i cluster per il tuo account nella AWS regione corrente.
3. Per visualizzare ulteriori dettagli di un cluster, scegli il nome del cluster nell'elenco.
4. In Actions (Operazioni), scegliere Manage IAM roles (Gestisci ruoli IAM). Verrà visualizzata la pagina Manage IAM roles (Gestisci ruoli IAM).
5. Aggiungere il ruolo IAM al cluster.

Esempio di utilizzo di una query federata

Negli esempi seguenti viene illustrato come eseguire una query federata. Esegui l'istruzione SQL utilizzando il client SQL connesso al database Amazon Redshift.

Esempio di utilizzo di una query federata con PostgreSQL

Nell'esempio seguente viene illustrato come configurare una query federata che faccia riferimento a un database Amazon Redshift, un database Aurora PostgreSQL e Amazon S3. Questo esempio illustra come funzionano le query federate. Per eseguirlo nel proprio ambiente, modificarlo in base al proprio ambiente. Per i prerequisiti per l'esecuzione, vedere Nozioni di base sull'utilizzo di query federate su PostgreSQL.

Creare uno schema esterno che faccia riferimento a un database Aurora PostgreSQL.

```
CREATE EXTERNAL SCHEMA apg
FROM POSTGRES
DATABASE 'database-1' SCHEMA 'myschema'
URI 'endpoint to aurora hostname'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-SecretsManager-RO'
```

```
SECRET_ARN 'arn:aws:secretsmanager:us-west-2:123456789012:secret:federation/test/
dataplane-apg-creds-YbVKQw';
```

Creare un altro schema esterno che faccia riferimento ad Amazon S3, che utilizza Amazon Redshift Spectrum. e concedere a `public` l'autorizzazione per utilizzare lo schema.

```
CREATE EXTERNAL SCHEMA s3
FROM DATA CATALOG
DATABASE 'default' REGION 'us-west-2'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-S3';

GRANT USAGE ON SCHEMA s3 TO public;
```

Mostrare il conteggio delle righe nella tabella Amazon Redshift.

```
SELECT count(*) FROM public.lineitem;

  count
----------
25075099
```

Mostrare il conteggio delle righe nella tabella Aurora PostgreSQL.

```
SELECT count(*) FROM apg.lineitem;

count
-------
11760
```

Mostrare il conteggio delle righe in Amazon S3.

```
SELECT count(*) FROM s3.lineitem_1t_part;

   count
------------
6144008876
```

Creare una vista delle tabelle da Amazon Redshift, Aurora PostgreSQL e Amazon S3. Questa visualizzazione viene utilizzata per eseguire la query federata.

```
CREATE VIEW lineitem_all AS
```

```
  SELECT
 l_orderkey,l_partkey,l_suppkey,l_linenumber,l_quantity,l_extendedprice,l_discount,l_tax,l_retu
        l_shipdate::date,l_commitdate::date,l_receiptdate::date,
 l_shipinstruct ,l_shipmode,l_comment
  FROM s3.lineitem_1t_part
  UNION ALL SELECT * FROM public.lineitem
  UNION ALL SELECT * FROM apg.lineitem
     with no schema binding;
```

Mostrare il conteggio delle righe nella visualizzazione `lineitem_all` con un predicato per limitare i risultati.

```
SELECT count(*) from lineitem_all WHERE l_quantity = 10;

   count
-----------
123373836
```

Scoprire quante vendite di un determinato articolo sono state effettuate nel mese di gennaio di ogni anno.

```
SELECT extract(year from l_shipdate) as year,
       extract(month from l_shipdate) as month,
       count(*) as orders
FROM lineitem_all
WHERE extract(month from l_shipdate) = 1
AND l_quantity < 2
GROUP BY 1,2
ORDER BY 1,2;

 year | month | orders
------+-------+---------
 1992 |     1 |  196019
 1993 |     1 | 1582034
 1994 |     1 | 1583181
 1995 |     1 | 1583919
 1996 |     1 | 1583622
 1997 |     1 | 1586541
 1998 |     1 | 1583198
 2016 |     1 |   15542
 2017 |     1 |   15414
```

```
 2018 |     1 |   15527
 2019 |     1 |     151
```

Esempio di utilizzo di un nome con formato maiuscole/minuscole misto

Per eseguire una query su un database remoto PostgreSQL supportato con un nome di un database, schema, tabella o colonna con lettere maiuscole e minuscole, impostare `enable_case_sensitive_identifier` su `true`. Per ulteriori informazioni su questo parametro di sessione, consultare enable_case_sensitive_identifier.

```
SET enable_case_sensitive_identifier TO TRUE;
```

In genere, i nomi di schemi e database sono in lettere minuscole. Nell'esempio seguente viene illustrato come connettersi a un database remoto PostgreSQL supportato con nomi minuscoli per database e schema e nomi con maiuscole e minuscole per tabella e colonna.

Creare uno schema esterno che faccia riferimento a un database Aurora PostgreSQL con un nome di database in minuscolo (`dblower`) e il nome dello schema in minuscolo (`schemalower`).

```
CREATE EXTERNAL SCHEMA apg_lower
FROM POSTGRES
DATABASE 'dblower' SCHEMA 'schemalower'
URI 'endpoint to aurora hostname'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-SecretsManager-RO'
SECRET_ARN 'arn:aws:secretsmanager:us-west-2:123456789012:secret:federation/test/
dataplane-apg-creds-YbVKQw';
```

Nella sessione in cui viene eseguita la query, impostare `enable_case_sensitive_identifier` su `true`.

```
SET enable_case_sensitive_identifier TO TRUE;
```

Eseguire una query federata per selezionare tutti i dati dal database PostgreSQL. La tabella (`MixedCaseTab`) e la colonna (`MixedCaseName`) hanno nomi con lettere maiuscole e minuscole. Il risultato è una riga (`Harry`).

```
select * from apg_lower."MixedCaseTab";
```

```
 MixedCaseName
-------
 Harry
```

Nell'esempio seguente viene illustrato come connettersi a un database remoto PostgreSQL supportato con nomi con lettere maiuscole e minuscole per database, schema, tabella e colonna.

Impostare `enable_case_sensitive_identifier` su `true` prima di creare lo schema esterno. Se `enable_case_sensitive_identifier` non è impostato su `true` prima della creazione dello schema esterno, si verifica un errore di database che non esiste.

Creare uno schema esterno che faccia riferimento a un database Aurora PostgreSQL con un nome di database (`UpperDB`) e di schema (`UpperSchema`) con lettere maiuscole e minuscole.

```
CREATE EXTERNAL SCHEMA apg_upper
FROM POSTGRES
DATABASE 'UpperDB' SCHEMA 'UpperSchema'
URI 'endpoint to aurora hostname'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-SecretsManager-RO'
SECRET_ARN 'arn:aws:secretsmanager:us-west-2:123456789012:secret:federation/test/
dataplane-apg-creds-YbVKQw';
```

Eseguire una query federata per selezionare tutti i dati dal database PostgreSQL. La tabella (`MixedCaseTab`) e la colonna (`MixedCaseName`) hanno nomi con lettere maiuscole e minuscole. Il risultato è una riga (`Harry`).

```
select * from apg_upper."MixedCaseTab";
```

```
 MixedCaseName
-------
 Harry
```

Esempio di utilizzo di una query federata con MySQL

Nell'esempio seguente viene illustrato come impostare una query federata che faccia riferimento a un database Aurora MySQL. Questo esempio illustra come funzionano le query federate. Per eseguirlo nel proprio ambiente, modificarlo in base al proprio ambiente. Per i prerequisiti per l'esecuzione, vedere Nozioni di base sull'utilizzo di query federate su MySQL (anteprima).

L'esempio dipende dai seguenti prerequisiti:

- Un segreto che è stato impostato in Secrets Manager per il database Aurora MySQL. A questo segreto viene fatto riferimento nelle policy di accesso e nei ruoli IAM. Per ulteriori informazioni, consultare Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate.
- Un gruppo di sicurezza impostato che collega Amazon Redshift e Aurora MySQL.

Creare uno schema esterno che faccia riferimento a un database Aurora MySQL.

```
CREATE EXTERNAL SCHEMA amysql
FROM MYSQL
DATABASE 'functional'
URI 'endpoint to remote hostname'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-SecretsManager-RO'
SECRET_ARN 'arn:aws:secretsmanager:us-west-2:123456789012:secret:federation/test/
dataplane-apg-creds-YbVKQw';
```

Eseguire un esempio di selezione SQL della tabella Aurora MySQL per visualizzare una riga dalla tabella dei dipendenti in Aurora MySQL.

```
SELECT level FROM amysql.employees LIMIT 1;

 level
-------
     8
```

Differenze dei tipi di dati tra Amazon Redshift e database PostgreSQL e MySQL supportati

Nella tabella seguente viene illustrata la mappatura di un tipo di dati di Amazon Redshift a un tipo di dati Amazon RDS PostgreSQL o Aurora PostgreSQL corrispondente.

Tipo di dati di Amazon Redshift	Tipo di dati RDS PostgreSQL o Aurora PostgreSQL	Descrizione
SMALLINT	SMALLINT	Intero a due byte firmato

Tipo di dati di Amazon Redshift	Tipo di dati RDS PostgreSQL o Aurora PostgreSQL	Descrizione
INTEGER	INTEGER	Intero a quattro byte firmato
BIGINT	BIGINT	Intero a otto byte firmato
DECIMAL	DECIMAL	Numerico esatto di precisione seleziona bile
REAL	REAL	Numero in virgola mobile a precisione singola
DOUBLE PRECISION	DOUBLE PRECISION	Numero in virgola mobile a precisione doppia
BOOLEAN	BOOLEAN	Booleani logici (true/ false)
CHAR	CHAR	Stringa di caratteri a lunghezza fissa
VARCHAR	VARCHAR	Stringa di caratteri a lunghezza variabile con un limite definito dall'utente
DATE	DATE	Data di calendario (anno, mese, giorno)
TIMESTAMP	TIMESTAMP	Data e ora (senza fuso orario)

Tipo di dati di Amazon Redshift	Tipo di dati RDS PostgreSQL o Aurora PostgreSQL	Descrizione
TIMESTAMPTZ	TIMESTAMPTZ	Data e ora (con fuso orario)
GEOMETRY	PostGIS GEOMETRY	Dati spaziali

I seguenti tipi di dati RDS PostgreSQL e Aurora PostgreSQL vengono convertiti in VARCHAR(64K) in Amazon Redshift:

- JSON, JSONB
- Matrici
- BIT, BIT VARYING
- BYTEA
- Tipi compositi
- Tipi di data e ora INTERVAL, TIME, TIME WITH TIMEZONE
- Tipi enumerati
- Tipi monetari
- Tipi di indirizzi di rete
- Tipi numerici SERIAL, BIGSERIAL, SMALLSERIAL e MONEY
- Tipi di identificatori di oggetti
- Tipo pg_lsn
- Pseudotipi
- Tipi di intervallo
- Tipi di ricerca testo
- TXID_SNAPSHOT
- UUID
- Tipo XML

Nella tabella seguente viene illustrata la mappatura di un tipo di dati Amazon Redshift a un tipo di dati Amazon RDS MySQL o Aurora MySQL.

Tipo di dati di Amazon Redshift	Tipo di dati RDS MySQL o Aurora MySQL	Descrizione
BOOLEAN	TINYINT(1)	Booleani logici (true o false)
SMALLINT	TINYINT(UNSIGNED)	Intero a due byte firmato
SMALLINT	SMALLINT	Intero a due byte firmato
INTEGER	SMALLINT UNSIGNED	Intero a quattro byte firmato
INTEGER	MEDIUMINT (UNSIGNED)	Intero a quattro byte firmato
INTEGER	INT	Intero a quattro byte firmato
BIGINT	INT UNSIGNED	Intero a otto byte firmato
BIGINT	BIGINT	Intero a otto byte firmato
DECIMAL	BIGINT UNSIGNED	Numerico esatto di precisione seleziona bile
DECIMAL	DECIMAL(M,D)	Numerico esatto di precisione seleziona bile
REAL	FLOAT	Numero in virgola mobile a precisione singola

Tipo di dati di Amazon Redshift	Tipo di dati RDS MySQL o Aurora MySQL	Descrizione
DOUBLE PRECISION	DOUBLE	Numero in virgola mobile a precisione doppia
CHAR	CHAR	Stringa di caratteri a lunghezza fissa
VARCHAR	VARCHAR	Stringa di caratteri a lunghezza variabile con un limite definito dall'utente
DATE	DATE	Data di calendario (anno, mese, giorno)
TIME	TIME	TIME (senza fuso orario)
TIMESTAMP	TIMESTAMP	Data e ora (senza fuso orario)
TIMESTAMP	DATETIME	TIME (senza fuso orario)
VARCHAR(4)	ANNO	Carattere a lunghezza variabile che rappresenta l'anno

Si verifica un errore quando i dati TIME sono fuori intervallo (00:00:00 - 24:00:00).

I seguenti tipi di dati RDS MySQL e Aurora MySQL vengono convertiti in VARCHAR(64K) in Amazon Redshift:

- BIT
- BINARY

- VARBINARY
- TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
- TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT
- ENUM
- SET
- SPATIAL

Considerazioni quando si accede ai dati federati con Amazon Redshift

Alcune funzionalità di Amazon Redshift non supportano l'accesso ai dati federati. Di seguito sono riportate le limitazioni e le considerazioni correlate.

Di seguito sono riportate limitazioni e considerazioni che si applicano quando si utilizzano query federate con Amazon Redshift:

- Le query federate supportano l'accesso in lettura a origini dati esterne. Non è possibile scrivere o creare oggetti database nell'origine dati esterna.
- In alcuni casi, potresti accedere a un database cluster Amazon RDS o Aurora DB in una regione AWS diversa da Amazon Redshift. In questi casi, in genere sono previsti costi di latenza di rete e di fatturazione per il trasferimento di dati tra regioni. AWS Ti consigliamo di utilizzare un database globale Aurora con un endpoint locale nella stessa AWS regione del cluster Amazon Redshift. I database globali Aurora utilizzano un'infrastruttura dedicata per la replica basata sullo storage in due regioni AWS con latenza tipica inferiore a 1 secondo.
- Considera il costo di accesso al cluster Amazon RDS o Aurora DB. Ad esempio, quando si utilizza questa funzionalità per accedere al cluster Aurora DB, i costi del cluster Aurora DB si basano sugli IOPS.
- Le query federate non consentono l'accesso ad Amazon Redshift dal cluster RDS o Aurora DB.
- Le query federate sono disponibili solo AWS nelle regioni in cui sono disponibili sia Amazon Redshift che Amazon RDS o il cluster Aurora DB.
- Le query federate al momento non supportano `ALTER SCHEMA`. Per modificare uno schema, utilizzare `DROP` e quindi `CREATE EXTERNAL SCHEMA`.
- Le query federate non funzionano con il dimensionamento simultaneo.

- Le query federate al momento non supportano l'accesso tramite un wrapper di dati esterni PostgreSQL.
- Le query federate per RDS MySQL o Aurora MySQL supportano l'isolamento delle transazioni a livello READ COMMITTED.
- Se non specificato, Amazon Redshift si connette a RDS per MySQL o ad Aurora MySQL sulla porta 3306. Verifica il numero di porta MySQL prima di creare uno schema esterno per MySQL.
- Se non specificato, Amazon Redshift si connette a RDS PostgreSQL o a PostgreSQL sulla porta 5432. Verifica il numero di porta PostgreSQL prima di creare uno schema esterno per PostgreSQL.
- Quando si recuperano i tipi di dati TIMESTAMP e DATE da MySQL, i valori zero vengono trattati come NULL.
- Se viene utilizzato un endpoint di lettura del database del cluster Aurora DB, può verificarsi un errore di «snapshot non valida». Ciò può essere evitato con uno dei seguenti metodi:
  - Utilizza un endpoint specifico dell'istanza del cluster Aurora DB (anziché utilizzare l'endpoint del cluster del cluster Aurora DB). Questo metodo utilizza l'isolamento delle transazioni REPEATABLE READ per i risultati dal database PostgreSQL.
  - Usa un endpoint Aurora DB Cluster Reader e imposta su false `pg_federation_repeatable_read` per la sessione. Questo metodo utilizza l'isolamento delle transazioni READ COMMITTED per i risultati dal database PostgreSQL. Per ulteriori informazioni sugli endpoint del lettore di cluster Aurora DB, consulta Tipi di endpoint del cluster Aurora DB nella Guida per l'utente di Amazon Aurora. Per informazioni su `pg_federation_repeatable_read`, consulta pg_federation_repeatable_read.

Di seguito sono riportate considerazioni per le transazioni quando si lavora con query federate su database PostgreSQL:

- Se una query è costituita da tabelle federate, il nodo direttivo avvia una transazione READ ONLY REPEATABLE READ nel database remoto. Questa transazione rimane per la durata della transazione Amazon Redshift.
- Il nodo direttivo crea uno snapshot del database remoto mediante chiamata `pg_export_snapshot` e crea un blocco di lettura sulle tabelle interessate.
- Un nodo di calcolo avvia una transazione e utilizza lo snapshot creato nel nodo direttivo per emettere query al database remoto.

Versioni supportate dei database federati

Uno schema esterno di Amazon Redshift fa riferimento a un database in un RDS PostgreSQL o Aurora PostgreSQL esterno. In tal caso, si applicano le seguenti limitazioni:

- Quando si crea uno schema esterno che fa riferimento al cluster Aurora DB, il database Aurora PostgreSQL deve avere la versione 9.6 o successiva.
- Quando si crea uno schema esterno che fa riferimento ad Amazon RDS, la versione del database PostgreSQL Amazon RDS deve essere 9.6 o successiva.

Uno schema esterno Amazon Redshift può fare riferimento a un database in un RDS MySQL o Aurora MySQL esterno. In tal caso, si applicano le seguenti limitazioni:

- Quando si crea uno schema esterno che fa riferimento al cluster Aurora DB, il database Aurora MySQL deve avere la versione 5.6 o successiva.
- Quando si crea uno schema esterno che fa riferimento ad Amazon RDS, la versione del database RDS MySQL deve essere 5.6 o successiva.

Esecuzione di query sui dati esterni utilizzando Amazon Redshift Spectrum

Mediante Amazon Redshift Spectrum, è possibile eseguire query e recuperare in modo efficace dati strutturati e semistrutturati dai file in Amazon S3 senza dover caricare i dati in tabelle Amazon Redshift. Le query di Redshift Spectrum vengono eseguite molto rapidamente su set di dati di grandi dimensioni grazie a un parallelismo massiccio. L'elaborazione viene eseguita principalmente nel livello di Redshift Spectrum e la maggior parte dei dati rimane in Amazon S3. Più cluster possono eseguire simultaneamente query sullo stesso set di dati in Amazon S3 senza la necessità di effettuare copie dei dati per ogni cluster.

Argomenti

Panoramica di Amazon Redshift Spectrum

Amazon Redshift Spectrum si trova su dei server Amazon Redshift dedicati indipendenti dal cluster. Amazon Redshift trasmette al livello Redshift Spectrum molte attività che richiedono un'importante capacità di calcolo, come l'aggregazione e il filtraggio di predicati. Le query di Redshift Spectrum

utilizzano quindi una capacità di elaborazione del cluster molto inferiore rispetto alle altre query. Redshift Spectrum consente inoltre un dimensionamento intelligente. In base alle richieste delle query, Redshift può potenzialmente utilizzare migliaia di istanze per beneficiare dell'elaborazione MPP (Massive Parallel Processing).

Per creare le tabelle di Redshift Spectrum, è necessario definire la struttura dei file e registrare quest'ultimi come tabelle in un catalogo dati esterno. Il catalogo dati esterno può essere AWS Glue il catalogo dati fornito con Amazon Athena o il tuo metastore Apache Hive. È possibile creare e gestire le tabelle esterne da Amazon Redshift utilizzando comandi DDL (data definition language) o qualsiasi altro strumento che si connette al catalogo di dati esterno. Le modifiche al catalogo di dati esterno sono immediatamente disponibili per tutti i cluster Amazon Redshift.

Inoltre, se lo desideri, puoi partizionare le tabelle esterne in una o più colonne. Questa operazione può consentire di migliorare le prestazioni, Il miglioramento si verifica in quanto l'ottimizzatore di query di Amazon Redshift elimina le partizioni che non contengono dati per la query.

Dopo la definizione delle tabelle di Redshift Spectrum, è possibile sottoporle a query e join esattamente come con qualunque altra tabella Amazon Redshift. Redshift Spectrum non supporta le operazioni di aggiornamento sulle tabelle esterne. Puoi aggiungere tabelle Redshift Spectrum a più cluster Amazon Redshift e interrogare gli stessi dati su Amazon S3 da qualsiasi cluster nella stessa regione. AWS Quando si aggiornano i file di dati Amazon S3, i dati diventano immediatamente disponibili per query da qualsiasi cluster Amazon Redshift.

Il catalogo AWS Glue dati a cui accedi potrebbe essere crittografato per aumentare la sicurezza. Se il AWS Glue catalogo è crittografato, è necessaria la chiave AWS Key Management Service (AWS KMS) AWS Glue per accedere al AWS Glue catalogo. AWS Glue la crittografia del catalogo non è disponibile in tutte le AWS regioni. Per un elenco delle AWS regioni supportate, consulta Encryption and Secure Access AWS Glue nella AWS Glue Developer Guide.Per ulteriori informazioni sulla crittografia del catalogo AWS Glue dati, consulta Encrypting Your AWS Glue Data Catalog nella Guida per gli AWS Glue sviluppatori.

> **Note**
>
> Non è possibile visualizzare i dettagli per le tabelle Redshift Spectrum che utilizzano le stesse risorse utilizzate per le tabelle Amazon Redshift standard come PG_TABLE_DEF, STV_TBL_PERM, PG_CLASS o information_schema. Se il tuo strumento di business intelligence o di analisi non riconosce le tabelle esterne Redshift Spectrum,

configura l'applicazione per eseguire la query su SVV_EXTERNAL_TABLES e SVV_EXTERNAL_COLUMNS.

Regioni di Amazon Redshift Spectrum

Redshift Spectrum è disponibile Regioni AWS dove è disponibile Amazon Redshift, se non diversamente specificato nella documentazione specifica della regione. Per Regione AWS la disponibilità nelle aree commerciali, consulta Endpoints di servizio per l'API Redshift nel. Riferimenti generali di Amazon Web Services

Considerazioni su Amazon Redshift Spectrum

Quando si utilizza Amazon Redshift Spectrum, tenere in considerazione quanto segue:

- Il cluster Amazon Redshift e il bucket Amazon S3 devono trovarsi nella stessa regione. AWS
- Redshift Spectrum non supporta il routing VPC avanzato con i cluster con provisioning. Per accedere ai dati di Amazon S3 potrebbe essere necessario eseguire fasi di configurazioni aggiuntive. Per ulteriori informazioni, consulta Utilizzo di Redshift Spectrum con routing VPC avanzato nella Guida alla gestione di Amazon Redshift.
- Redshift Spectrum supporta gli alias degli Access Point Amazon S3. Per ulteriori informazioni, consulta Utilizzo di un alias in stile bucket per il punto di accesso nella Guida dell'utente Amazon Simple Storage Service. Tuttavia, Redshift Spectrum non supporta VPC con alias dei punti di accesso Amazon S3. Per ulteriori informazioni, consulta Utilizzo di Redshift Spectrum con routing VPC avanzato nella Guida alla gestione di Amazon Redshift.
- Non puoi eseguire operazioni di aggiornamento o eliminazione sulle tabelle esterne. Per creare una nuova tabella esterna nello schema specificato, puoi utilizzare CREATE EXTERNAL TABLE. Per ulteriori informazioni su CREATE EXTERNAL TABLE AS, consultare CREATE EXTERNAL TABLE. Per inserire i risultati di una query SELECT nelle tabelle esterne esistenti nei cataloghi esterni, puoi utilizzare INSERT (tabella esterna). Per ulteriori informazioni su INSERT (tabella esterna), consultare INSERT (tabella esterna).
- A meno AWS Glue Data Catalog che tu non stia utilizzando uno abilitato per AWS Lake Formation, non puoi controllare le autorizzazioni degli utenti su una tabella esterna. Puoi invece concedere e revocare autorizzazioni per lo schema esterno. Per ulteriori informazioni su come lavorare con AWS Lake Formation, vedereUtilizzo di Redshift Spectrum con AWS Lake Formation.

- Per eseguire le query di Redshift Spectrum, l'utente del database deve disporre dell'autorizzazione per creare tabelle temporanee nel database. L'esempio seguente concede l'autorizzazione temporanea per il database `spectrumdb` al gruppo di utenti `spectrumusers`.

  ```
  grant temp on database spectrumdb to group spectrumusers;
  ```

  Per ulteriori informazioni, consultare GRANT.
- Quando utilizzi Athena Data Catalog o AWS Glue Data Catalog come archivio di metadati, consulta Quotas and Limits nella Amazon Redshift Management Guide.
- Redshift Spectrum non supporta Amazon EMR con Kerberos.

Nozioni di base su Amazon Redshift Spectrum

In questo tutorial viene descritto come utilizzare Amazon Redshift Spectrum per eseguire le query sui dati direttamente dai file in Amazon S3. Se disponi già di un cluster e di un client SQL, puoi completare questo tutorial con un intervento minimo di impostazione.

> **Note**
>
> Le query di Redshift Spectrum comportano dei costi supplementari. Quello relativo all'esecuzione delle query di esempio in questo tutorial è nominale. Per ulteriori informazioni sui prezzi, consulta Prezzi di Amazon Redshift Spectrum.

Prerequisiti

Per utilizzare Redshift Spectrum, è necessario disporre di un cluster Amazon Redshift e un client SQL connesso al cluster di modo che sia possibile eseguire i comandi SQL. Il cluster e i file di dati in Amazon S3 devono trovarsi nella stessa Regione AWS.

Per informazioni su come creare un cluster Amazon Redshift, consulta Amazon Redshift provisioned clusters nella Amazon Redshift Getting Started Guide. Per informazioni sulle modalità di connessione a un cluster, consulta Connecting to Amazon Redshift data warehouse nella Amazon Redshift Getting Started Guide.

In alcuni degli esempi che seguono, i dati di esempio si trovano nella regione Stati Uniti orientali (Virginia settentrionale) (`us-east-1`), quindi è necessario un cluster che si trova in `us-east-1`. In

alternativa, puoi utilizzare Amazon S3 per copiare oggetti di dati dai seguenti bucket e cartelle nel tuo bucket nel luogo in Regione AWS cui si trova il cluster:

- `s3://redshift-downloads/tickit/spectrum/customers/*`
- `s3://redshift-downloads/tickit/spectrum/sales_partition/*`
- `s3://redshift-downloads/tickit/spectrum/sales/*`
- `s3://redshift-downloads/tickit/spectrum/salesevent/*`

Esegui un comando Amazon S3 simile al seguente per copiare i dati di esempio che si trovano negli Stati Uniti orientali (Virginia settentrionale) nel tua Regione AWS. Prima di eseguire il comando, crea il bucket e le cartelle nel bucket in modo che corrispondano al comando di copia di Amazon S3. L'output del comando di copia di Amazon S3 conferma che i file vengono copiati nel *nome_bucket* nella Regione AWS desiderata.

```
aws s3 cp s3://redshift-downloads/tickit/spectrum/ s3://bucket-name/tickit/spectrum/ --
copy-props none --recursive
```

Guida introduttiva a Redshift Spectrum con AWS CloudFormation

In alternativa ai seguenti passaggi, puoi accedere al DataLake AWS CloudFormation modello Redshift Spectrum per creare uno stack con un bucket Amazon S3 su cui eseguire query. Per ulteriori informazioni, consulta Avvia lo AWS CloudFormation stack e quindi interroga i dati in Amazon S3.

Nozioni di base su Redshift Spectrum graduale

Per iniziare a utilizzare Amazon Redshift Spectrum, completare la seguente procedura:

- Fase 1: Creazione di un ruolo IAM per Amazon Redshift
- Fase 2: associazione del ruolo IAM al cluster
- Fase 3: creazione di uno schema esterno e di una tabella esterna
- Fase 4: Esecuzione di query sui dati in Amazon S3

Fase 1: Creazione di un ruolo IAM per Amazon Redshift

Il cluster necessita dell'autorizzazione per accedere al catalogo dati esterno in AWS Glue Amazon Athena e ai file di dati in Amazon S3. Per concedere questa autorizzazione, fare riferimento a un

ruolo AWS Identity and Access Management (IAM) associato al cluster. Per ulteriori informazioni sull'utilizzo di ruoli con Amazon Redshift, consultare Autorizzazione delle operazioni COPY e UNLOAD tramite ruoli IAM.

> **Note**
>
> In alcuni casi, puoi migrare il tuo Athena Data Catalog a AWS Glue un Data Catalog. Puoi farlo se il tuo cluster si trova in una AWS regione in cui AWS Glue è supportato e hai tabelle esterne Redshift Spectrum nell'Athena Data Catalog. Per utilizzare AWS Glue Data Catalog con Redshift Spectrum, potrebbe essere necessario modificare le policy IAM. Per ulteriori informazioni, consultare Aggiornamento del catalogo dati AWS Glue nella Guida per l'utente di Athena.

Quando viene creato un ruolo per Amazon Redshift, scegliere uno dei seguenti approcci:

- Se utilizzi Redshift Spectrum con un Athena Data Catalog o AWS Glue Data Catalog, segui i passaggi descritti in. Come creare un ruolo IAM per Amazon Redshift
- Se utilizzi Redshift Spectrum con un AWS Glue Data Catalog programma abilitato per AWS Lake Formation, segui i passaggi descritti nelle seguenti procedure:
  - Per creare un ruolo IAM per Amazon Redshift utilizzando un AWS Glue Data Catalog enabled for AWS Lake Formation
  - Come concedere le autorizzazioni SELECT nella tabella per eseguire le query del database Lake Formation

Come creare un ruolo IAM per Amazon Redshift

1. Aprire la console IAM.
2. Nel pannello di navigazione, selezionare Ruoli.
3. Selezionare Create role (Crea ruolo).
4. Scegliere Servizio AWS come entità attendibile, quindi scegliere Redshift come caso d'uso.
5. In Caso d'uso per altro Servizi AWS, scegli Redshift - Personalizzabile, quindi scegli Avanti.
6. Verrà visualizzata la pagina Allega la policy di autorizzazione. Scegli `AmazonS3ReadOnlyAccess` e`AWSGlueConsoleFullAccess`, se utilizzi il AWS Glue Data Catalog. Oppure scegliere `AmazonAthenaFullAccess` se si utilizza il catalogo dati di Athena. Seleziona Successivo.

> **Note**
>
> La policy `AmazonS3ReadOnlyAccess` concede al cluster l'accesso in sola lettura a tutti i bucket Amazon S3. Per concedere l'accesso solo al bucket di dati di AWS esempio, crea una nuova politica e aggiungi le seguenti autorizzazioni.
>
> ```
> {
>     "Version": "2012-10-17",
>     "Statement": [
>         {
>             "Effect": "Allow",
>             "Action": [
>                 "s3:Get*",
>                 "s3:List*"
>             ],
>             "Resource": "arn:aws:s3:::redshift-downloads/*"
>         }
>     ]
> }
> ```

7. Per Role name (Nome ruolo), digitare un nome per il ruolo, ad esempio **`myspectrum_role`**.
8. Esaminare le informazioni, quindi scegliere Create role (Crea ruolo).
9. Nel riquadro di navigazione, seleziona Ruoli. Scegliere il nome del nuovo ruolo per visualizzare il riepilogo, quindi copiare il ruolo ARN visualizzato nel campo Role ARN (ARN ruolo). Questo valore è l'Amazon Resource Name (ARN) per il ruolo appena creato. Viene utilizzato quando si creano tabelle esterne per fare riferimento ai file di dati in Amazon S3.

Per creare un ruolo IAM per Amazon Redshift utilizzando un AWS Glue Data Catalog enabled for AWS Lake Formation

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, selezionare Policy.

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.
3. Scegli Crea policy.
4. Scegliere di creare la policy nella scheda JSON.

5. Incollare nel documento di policy JSON seguente, che concede l'accesso al catalogo di dati ma nega le autorizzazioni di amministratore per Lake Formation.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RedshiftPolicyForLF",
            "Effect": "Allow",
            "Action": [
                "glue:*",
                "lakeformation:GetDataAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

6. Una volta terminato, selezionare Review (Rivedi) per rivedere la policy. In Validatore di policy vengono segnalati eventuali errori di sintassi.
7. Nella pagina Review policy (Rivedi policy), in Name (Nome) inserire **`myspectrum_policy`** per denominare la policy in fase di creazione. (Opzionale) Immettere una Description (descrizione). Consulta il Summary (Riepilogo) della policy per visualizzare le autorizzazioni concesse dalla policy. Seleziona Create policy (Crea policy) per salvare il proprio lavoro.

   Dopo aver creato la policy, è possibile fornire l'accesso agli utenti.

Per fornire l'accesso, aggiungi autorizzazioni ai tuoi utenti, gruppi o ruoli:

- Utenti e gruppi in AWS IAM Identity Center:

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center .
- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Creating a role for a third-party identity provider (federation) (Creazione di un ruolo per un provider di identità di terze parti [federazione]) nella Guida per l'utente di IAM.
- Utenti IAM:

- Crea un ruolo che l'utente possa assumere. Per istruzioni, consulta la pagina Creating a role for an IAM user (Creazione di un ruolo per un utente IAM) nella Guida per l'utente di IAM.
- (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente di IAM.

Come concedere le autorizzazioni SELECT nella tabella per eseguire le query del database Lake Formation

1. Aprire la console Lake Formation all'indirizzo https://console.aws.amazon.com/lakeformation/.
2. Nel menu di navigazione scegli Autorizzazioni data lake, quindi seleziona Concedi.
3. Segui le istruzioni presenti in Granting table permissions using the named resource method nella Guida per gli sviluppatori di AWS Lake Formation . Inserisci le informazioni che seguono:
   - In Ruolo IAM, selezionare il ruolo IAM creato, `myspectrum_role`. Quando si esegue l'editori di query di Amazon Redshift, utilizzare il ruolo IAM per le autorizzazioni ai dati.

     Note

     Per concedere l'autorizzazione SELECT nella tabella in un catalogo di dati abilitato per Lake Formation eseguire la query, procedere come segue:

     - Registrare il percorso dei dati in Lake Formation.
     - Concedere agli utenti le autorizzazioni per quel percorso in Lake Formation.
     - Le tabelle create sono disponibili nel percorso registrato in Lake Formation.
4. Scegli Concessione.

Important

Come best practice, concedere l'accesso solo agli oggetti Amazon S3 sottostanti attraverso le autorizzazioni Lake Formation. Per evitare un accesso non approvato, rimuovere qualsiasi autorizzazione concessa agli oggetti Amazon S3 al di fuori di Lake Formation. Se l'accesso agli oggetti Amazon S3 è stato effettuato prima della configurazione di Lake Formation, rimuovere qualsiasi autorizzazione di policy IAM o bucket configurata in precedenza. Per

ulteriori informazioni, consulta Upgrading AWS Glue Data Permissions to the AWS Lake Formation Model e Lake Formation Permissions.

Fase 2: associazione del ruolo IAM al cluster

Ora si dispone di un ruolo IAM che autorizza Amazon Redshift ad accedere al catalogo dati esterno e ad Amazon S3 per tuo conto. A questo punto è necessario associare tale ruolo al proprio cluster Amazon Redshift.

Associare un ruolo IAM a un cluster

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione scegliere Clusters (Cluster), quindi scegliere il cluster da aggiornare.
3. In Actions (Operazioni), scegliere Manage IAM roles (Gestisci ruoli IAM). Viene visualizzata la pagina IAM roles (Ruoli IAM).
4. Scegliere Inserisci ARN e quindi inserire un ARN o un ruolo IAM, oppure scegliere un ruolo IAM dall'elenco. Quindi scegliere Add IAM role (Aggiungi ruolo IAM) per aggiungerlo all'elenco degli Attached IAM roles (Ruoli IAM collegati).
5. Scegliere Done (Fatto) per associare il ruolo IAM al cluster. Il cluster viene modificare per completare la variazione.

Fase 3: creazione di uno schema esterno e di una tabella esterna

Creazione di tabelle esterne in uno schema esterno. Lo schema esterno fa riferimento a un database nel catalogo dati esterno e fornisce l'ARN del ruolo IAM che autorizza il tuo cluster ad accedere ad Amazon S3 per tuo conto. Puoi creare un database esterno in un Amazon Athena Data Catalog o in un metastore Apache Hive AWS Glue Data Catalog, come Amazon EMR. Per questo esempio, viene creato un database esterno in un catalogo dati Amazon Athena quando viene creato lo schema esterno di Amazon Redshift. Per ulteriori informazioni, consultare Creazione di schemi esterni per Amazon Redshift Spectrum.

Per creare uno schema esterno e una tabella esterna

1. Per creare uno schema esterno, sostituire l'ARN del ruolo IAM nel comando seguente con l'ARN del ruolo creato nella fase 1. Quindi eseguire il comando nel proprio client SQL.

```
create external schema myspectrum_schema
from data catalog
database 'myspectrum_db'
iam_role 'arn:aws:iam::123456789012:role/myspectrum_role'
create external database if not exists;
```

2. Per creare una tabella esterna, eseguire il comando CREATE EXTERNAL TABLE seguente.

> **Note**
>
> Il cluster e il bucket Amazon S3 devono trovarsi nella stessa Regione AWS. Per questo esempio del comando CREATE EXTERNAL TABLE, il bucket Amazon S3 con i dati di esempio si trova negli Stati Uniti orientali (Virginia settentrionale). Regione AWS Per visualizzare i dati di origine, scarica il file `sales_ts.000`. .
> È possibile modificare questo esempio per eseguirlo in un altro Regione AWS. Crea un bucket Amazon S3 nel formato desiderato. Regione AWS Copia i dati di vendita con un comando di copia di Amazon S3. Aggiorna quindi l'opzione relativa alla posizione del bucket nel comando di esempio `CREATE EXTERNAL TABLE` impostando il bucket in uso.
>
> ```
> aws s3 cp s3://redshift-downloads/tickit/spectrum/sales/ s3://bucket-name/
> tickit/spectrum/sales/ --copy-props none --recursive
> ```
>
> L'output del comando di copia di Amazon S3 conferma che il file è stato copiato nel *bucket-name* nella Regione AWS desiderata.
>
> ```
> copy: s3://redshift-downloads/tickit/spectrum/sales/sales_ts.000 to
>  s3://bucket-name/tickit/spectrum/sales/sales_ts.000
> ```

```
create external table myspectrum_schema.sales(
salesid integer,
listid integer,
sellerid integer,
buyerid integer,
eventid integer,
dateid smallint,
qtysold smallint,
```

```
pricepaid decimal(8,2),
commission decimal(8,2),
saletime timestamp)
row format delimited
fields terminated by '\t'
stored as textfile
location 's3://redshift-downloads/tickit/spectrum/sales/'
table properties ('numRows'='172000');
```

Fase 4: Esecuzione di query sui dati in Amazon S3

Dopo aver creato le tabelle esterne, è possibile eseguire query utilizzando le stesse istruzioni SELECT utilizzate per eseguire query su altre tabelle Amazon Redshift. Queste query con istruzioni SELECT includono il join di tabelle, l'aggregazione di dati e il filtraggio di predicati.

Come eseguire query sui dati in Amazon S3

1. Ottenere Il numero di righe nella tabella MYSPECTRUM_SCHEMA.SALES.

   ```
   select count(*) from myspectrum_schema.sales;
   ```

   ```
   count
   ------
   172462
   ```

2. Come best practice, lasciare le tabelle dei fatti più grandi in Amazon S3 e le tabelle con dimensioni più piccole in Amazon Redshift. Se hai caricato i dati di esempio in Load data, hai una tabella denominata EVENT nel tuo database. Altrimenti, creare la tabella EVENT utilizzando il comando seguente.

   ```
   create table event(
   eventid integer not null distkey,
   venueid smallint not null,
   catid smallint not null,
   dateid smallint not null sortkey,
   eventname varchar(200),
   starttime timestamp);
   ```

3. Caricare la tabella EVENT sostituendo l'ARN del ruolo IAM nel seguente comando COPY con l'ARN del ruolo creato in Fase 1: Creazione di un ruolo IAM per Amazon Redshift.

Facoltativamente, puoi scaricare e visualizzare i dati di origine `allevents_pipe.txt` da un bucket Amazon S3. Regione AWS `us-east-1`

```
copy event from 's3://redshift-downloads/tickit/allevents_pipe.txt'
iam_role 'arn:aws:iam::123456789012:role/myspectrum_role'
delimiter '|' timeformat 'YYYY-MM-DD HH:MI:SS' region 'us-east-1';
```

L'esempio seguente unisce mediante join la tabella esterna Amazon S3 MYSPECTRUM_SCHEMA.SALES alla tabella Amazon Redshift locale EVENT per calcolare le vendite totali per i 10 eventi principali.

```
select top 10 myspectrum_schema.sales.eventid,
 sum(myspectrum_schema.sales.pricepaid) from myspectrum_schema.sales, event
where myspectrum_schema.sales.eventid = event.eventid
and myspectrum_schema.sales.pricepaid > 30
group by myspectrum_schema.sales.eventid
order by 2 desc;
```

```
eventid | sum
--------+---------
    289 | 51846.00
   7895 | 51049.00
   1602 | 50301.00
    851 | 49956.00
   7315 | 49823.00
   6471 | 47997.00
   2118 | 47863.00
    984 | 46780.00
   7851 | 46661.00
   5638 | 46280.00
```

4. Visualizza il piano delle query per la query precedente. Tenere presente le fasi `S3 Seq Scan`, `S3 HashAggregate` e `S3 Query Scan` eseguite sui dati in Amazon S3.

```
explain
select top 10 myspectrum_schema.sales.eventid,
 sum(myspectrum_schema.sales.pricepaid)
from myspectrum_schema.sales, event
where myspectrum_schema.sales.eventid = event.eventid
and myspectrum_schema.sales.pricepaid > 30
group by myspectrum_schema.sales.eventid
```

```
order by 2 desc;
```

```
QUERY PLAN

------------------------------------------------------------------------------
XN Limit  (cost=1001055770628.63..1001055770628.65 rows=10 width=31)

  ->  XN Merge  (cost=1001055770628.63..1001055770629.13 rows=200 width=31)

        Merge Key: sum(sales.derived_col2)

        ->  XN Network  (cost=1001055770628.63..1001055770629.13 rows=200 width=31)

              Send to leader

              ->  XN Sort  (cost=1001055770628.63..1001055770629.13 rows=200
 width=31)

                    Sort Key: sum(sales.derived_col2)

                    ->  XN HashAggregate  (cost=1055770620.49..1055770620.99
 rows=200 width=31)

                          ->  XN Hash Join DS_BCAST_INNER
 (cost=3119.97..1055769620.49 rows=200000 width=31)

                                Hash Cond: ("outer".derived_col1 = "inner".eventid)

                                ->  XN S3 Query Scan sales  (cost=3010.00..5010.50
 rows=200000 width=31)

                                      ->  S3 HashAggregate  (cost=3010.00..3010.50
 rows=200000 width=16)
```

```
                                          ->  S3 Seq Scan myspectrum_schema.sales
location:"s3://redshift-downloads/tickit/spectrum/sales" format:TEXT
(cost=0.00..2150.00 rows=172000 width=16)
                                                Filter: (pricepaid > 30.00)


                               ->  XN Hash  (cost=87.98..87.98 rows=8798 width=4)


                                     ->  XN Seq Scan on event  (cost=0.00..87.98
rows=8798 width=4)
```

Avvia lo AWS CloudFormation stack e quindi interroga i dati in Amazon S3

Dopo aver creato un cluster Amazon Redshift e esserti connesso al cluster, puoi installare il DataLake AWS CloudFormation modello Redshift Spectrum e quindi interrogare i tuoi dati.

CloudFormation installa il modello Redshift Spectrum Getting DataLake Started e crea uno stack che include quanto segue:

- Un ruolo denominato `myspectrum_role` associato al cluster Redshift
- Uno schema esterno denominato `myspectrum_schema`
- Una tabella esterna denominata `sales` in un bucket Amazon S3
- Una tabella Redshift denominata `event` caricata con dati

Per avviare lo stack Redshift Spectrum Getting Started DataLake CloudFormation

1. Scegliere Avvio dello stack CFN. La CloudFormation console si apre con il template DataLake .yml selezionato.

   Puoi anche scaricare e personalizzare il modello DataLake CloudFormation CFN Redshift Spectrum Getting Started, quindi aprire la CloudFormation console (https://console.aws.amazon.com/cloudformation) e creare uno stack con il modello personalizzato.
2. Seleziona Successivo.
3. In Parametri, inserisci il nome del cluster Amazon Redshift, il nome del database e il nome utente del database.
4. Seleziona Successivo.

Vengono visualizzate le opzioni della pila.

5. Scegliere Successivo per accettare le impostazioni predefinite.
6. Consulta le informazioni e nella sezione Capacità, quindi scegli Riconosco che AWS CloudFormation potrebbe creare risorse IAM.
7. Seleziona Crea stack.

Se si verifica un errore durante la creazione della pila, vedere le seguenti informazioni:

- Visualizza la scheda CloudFormation Eventi per informazioni che possono aiutarti a risolvere l'errore.
- Elimina lo DataLake CloudFormation stack prima di riprovare l'operazione.
- Assicurarsi di essere connesso al database Amazon Redshift.
- Assicurati di aver inserito le informazioni corrette per il nome del cluster Amazon Redshift, il nome del database e il nome utente del database.

Eseguire query sui dati in Amazon S3

Richiedere tabelle esterne utilizzando le stesse istruzioni SELECT utilizzate per eseguire query su altre tabelle Amazon Redshift. Queste query con istruzioni SELECT includono il join di tabelle, l'aggregazione di dati e il filtraggio di predicati.

La seguente query restituisce il numero di righe nella tabella esternamyspectrum_schema.sales.

```
select count(*) from myspectrum_schema.sales;
```

```
count
------
172462
```

Unire una tabella esterna a una locale

L'esempio seguente unisce la tabella esterna myspectrum_schema.sales alla tabella locale event per calcolare le vendite totali per i 10 eventi principali.

```
select top 10 myspectrum_schema.sales.eventid, sum(myspectrum_schema.sales.pricepaid)
 from myspectrum_schema.sales, event
```

```
where myspectrum_schema.sales.eventid = event.eventid
and myspectrum_schema.sales.pricepaid > 30
group by myspectrum_schema.sales.eventid
order by 2 desc;
```

```
eventid | sum
--------+---------
    289 | 51846.00
   7895 | 51049.00
   1602 | 50301.00
    851 | 49956.00
   7315 | 49823.00
   6471 | 47997.00
   2118 | 47863.00
    984 | 46780.00
   7851 | 46661.00
   5638 | 46280.00
```

Visualizzazione del piano di query

Visualizza il piano delle query per la query precedente. Tenere presente le fasi `S3 Seq Scan`, `S3 HashAggregate` e `S3 Query Scan` eseguite nei dati in Amazon S3.

```
explain
select top 10 myspectrum_schema.sales.eventid, sum(myspectrum_schema.sales.pricepaid)
from myspectrum_schema.sales, event
where myspectrum_schema.sales.eventid = event.eventid
and myspectrum_schema.sales.pricepaid > 30
group by myspectrum_schema.sales.eventid
order by 2 desc;
```

```
QUERY PLAN


------------------------------------------------------------------------------
XN Limit  (cost=1001055770628.63..1001055770628.65 rows=10 width=31)


  ->  XN Merge  (cost=1001055770628.63..1001055770629.13 rows=200 width=31)
```

```
        Merge Key: sum(sales.derived_col2)

        ->  XN Network  (cost=1001055770628.63..1001055770629.13 rows=200 width=31)

              Send to leader

              ->  XN Sort  (cost=1001055770628.63..1001055770629.13 rows=200 width=31)

                    Sort Key: sum(sales.derived_col2)

                    ->  XN HashAggregate  (cost=1055770620.49..1055770620.99 rows=200
width=31)

                          ->  XN Hash Join DS_BCAST_INNER  (cost=3119.97..1055769620.49
rows=200000 width=31)

                                Hash Cond: ("outer".derived_col1 = "inner".eventid)

                                ->  XN S3 Query Scan sales  (cost=3010.00..5010.50
rows=200000 width=31)

                                      ->  S3 HashAggregate  (cost=3010.00..3010.50
rows=200000 width=16)

                                            ->  S3 Seq Scan spectrum.sales
location:"s3://redshift-downloads/tickit/spectrum/sales" format:TEXT
(cost=0.00..2150.00 rows=172000 width=16)
                                                  Filter: (pricepaid > 30.00)

                                ->  XN Hash  (cost=87.98..87.98 rows=8798 width=4)

                                      ->  XN Seq Scan on event  (cost=0.00..87.98
rows=8798 width=4)
```

Policy IAM per Amazon Redshift Spectrum

Per impostazione predefinita, Amazon Redshift Spectrum utilizza AWS Glue Data Catalog le AWS regioni che AWS Glue supportano. In altre AWS regioni, Redshift Spectrum utilizza il catalogo dati Athena. Il tuo cluster necessita dell'autorizzazione per accedere al tuo catalogo di dati esterno in AWS Glue o Athena e ai tuoi file di dati in Amazon S3. Fornisci tale autorizzazione facendo riferimento a un ruolo AWS Identity and Access Management (IAM) collegato al tuo cluster. Se si utilizza un metastore Apache Hive per gestire il catalogo dati, non sarà necessario fornire l'accesso a ad Athena.

Puoi concatenare i ruoli di modo che il cluster possa assumere altri ruoli non associati al cluster. Per ulteriori informazioni, consulta Concatenazione di ruoli IAM per Amazon Redshift Spectrum.

Il AWS Glue catalogo a cui accedi potrebbe essere crittografato per aumentare la sicurezza. Se il AWS Glue catalogo è crittografato, è necessaria la AWS KMS chiave AWS Glue per accedere al catalogo AWS Glue dati. Per ulteriori informazioni, consulta Encrypting Your AWS Glue Data Catalog nella AWS Glue Developer Guide.

Argomenti

Autorizzazioni di Amazon S3

Il cluster deve disporre almeno dell'accesso GET e LIST per il bucket Amazon S3. Se il bucket non si trova nello stesso AWS account del cluster, il bucket deve inoltre autorizzare il cluster ad accedere ai dati. Per ulteriori informazioni, consulta Autorizzazione di Amazon Redshift ad accedere ad AWS altri servizi per tuo conto.

> **Note**
>
> Il bucket Amazon S3 non può utilizzare una policy di bucket che limita l'accesso solo da specifici endpoint VPC.

La policy seguente concede l'accesso GET e LIST a qualsiasi bucket Amazon S3. La policy consente l'accesso a bucket Amazon S3 per Redshift Spectrum nonché le operazioni COPY.

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": ["s3:Get*", "s3:List*"],
  "Resource": "*"
 }]
}
```

La policy seguente concede l'accesso GET e LIST al bucket Amazon S3 denominato `myBucket`.

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": ["s3:Get*", "s3:List*"],
  "Resource": "arn:aws:s3:::myBucket/*"
 }]
}
```

Autorizzazioni Amazon S3 tra account

Per concedere a Redshift Spectrum l'autorizzazione ad accedere ai dati in un bucket Amazon S3 che appartiene a AWS un altro account, aggiungi la seguente policy al bucket Amazon S3. Per ulteriori informazioni, consultare la pagina relativa alla concessione di autorizzazioni per bucket multiaccount.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Example permissions",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::redshift-account:role/spectrumrole"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListMultipartUploadParts",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::bucketname",
                "arn:aws:s3:::bucketname/*"
            ]
        }
    ]
}
```

Policy per concedere o limitare l'accesso mediante Redshift Spectrum

Per concedere l'accesso a un bucket Amazon S3 solo mediante Redshift Spectrum, includere una condizione che consenta l'accesso per l'agente utente `AWS Redshift/Spectrum`. La policy seguente consente l'accesso a bucket Amazon S3 unicamente per Redshift Spectrum. Esclude gli altri accessi, come le operazioni COPY.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": ["s3:Get*", "s3:List*"],
        "Resource": "arn:aws:s3:::myBucket/*",
                "Condition": {"StringEquals": {"aws:UserAgent": "AWS Redshift/
Spectrum"}}
    }]
}
```

Analogamente, è possibile creare un ruolo IAM che consente l'accesso alle operazioni COPY, ma esclude l'accesso a Redshift Spectrum. A questo proposito, è necessario includere una condizione che nega l'accesso per l'agente utente **`AWS Redshift/Spectrum`**. La policy seguente consente l'accesso a un bucket Amazon S3 a eccezione di Redshift Spectrum.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": ["s3:Get*", "s3:List*"],
        "Resource": "arn:aws:s3:::myBucket/*",
                "Condition": {"StringNotEquals": {"aws:UserAgent": "AWS Redshift/
Spectrum"}}
    }]
}
```

Policy per concedere autorizzazioni minime

La seguente politica concede le autorizzazioni minime richieste per utilizzare Redshift Spectrum con Amazon S3 e Athena. AWS Glue

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListMultipartUploadParts",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::bucketname",
                "arn:aws:s3:::bucketname/folder1/folder2/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:UpdateDatabase",
                "glue:CreateTable",
```

```
                "glue:DeleteTable",
                "glue:BatchDeleteTable",
                "glue:UpdateTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:BatchCreatePartition",
                "glue:CreatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:UpdatePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Se invece utilizzi Athena per il tuo catalogo dati AWS Glue, la policy richiede l'accesso completo ad Athena. La policy seguente concede l'accesso alle risorse Athena. Se il database esterno è un metastore Hive, l'accesso ad Athena non è necessario.

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": ["athena:*"],
  "Resource": ["*"]
 }]
}
```

Concatenazione di ruoli IAM per Amazon Redshift Spectrum

Quando associ un ruolo al cluster, quest'ultimo può assumere quel ruolo per accedere ad Amazon S3, Athena e AWS Glue per tuo conto. Se un ruolo collegato al cluster non ha accesso alle risorse necessarie, è possibile concatenare un altro ruolo, possibilmente appartenente a un altro account. Il cluster quindi può assumere temporaneamente il ruolo concatenato per accedere ai dati. Concatenando i ruoli è anche possibile concedere l'accesso a più account. È possibile concatenare

un massimo di 10 ruoli. Ogni ruolo nella catena assume il ruolo successivo nella catena, fino a quando il cluster non assume il ruolo alla fine della catena.

Per concatenare i ruoli, stabilisci una relazione di trust tra di essi. Un ruolo che assume un altro ruolo deve avere una policy di autorizzazioni che gli consente di assumere il ruolo specificato. A sua volta, il ruolo che passa le autorizzazioni deve avere una policy di trust che gli consente di passare le relative autorizzazioni a un altro ruolo. Per ulteriori informazioni, consultare Concatenazione di ruoli IAM in Amazon Redshift.

Quando esegui il comando CREATE EXTERNAL SCHEMA, puoi concatenare i ruoli includendo un elenco di ARN di ruoli separati da virgole.

> **Note**
>
> L'elenco di ruoli concatenati non deve includere spazi.

Negli esempi seguenti, `MyRedshiftRole` è collegato al cluster. `MyRedshiftRole` assume il ruolo `AcmeData`, che appartiene all'account `111122223333`.

```
create external schema acme from data catalog
database 'acmedb' region 'us-west-2'
iam_role 'arn:aws:iam::123456789012:role/MyRedshiftRole,arn:aws:iam::111122223333:role/
AcmeData';
```

Controllo dell'accesso al Data Catalog AWS Glue

Se lo utilizzi AWS Glue per il tuo catalogo di dati, puoi applicare un controllo granulare degli accessi al AWS Glue Data Catalog con la tua policy IAM. Ad esempio, potresti decidere di esporre soltanto alcuni database e tabelle a uno specifico ruolo IAM.

Le sezioni seguenti descrivono le politiche IAM per i vari livelli di accesso ai dati archiviati nel AWS Glue Data Catalog.

Argomenti

- Policy per le operazioni di database
- Policy per le operazioni di tabella
- Policy per le operazioni di partizione

Policy per le operazioni di database

Se desideri concedere agli utenti le autorizzazioni per visualizzare e creare un database, devono disporre dei diritti di accesso sia al database che al AWS Glue Data Catalog.

Nella seguente query di esempio viene creato un database.

```
CREATE EXTERNAL SCHEMA example_db
FROM DATA CATALOG DATABASE 'example_db' region 'us-west-2'
IAM_ROLE 'arn:aws:iam::redshift-account:role/spectrumrole'
CREATE EXTERNAL DATABASE IF NOT EXISTS
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per la creazione di un database.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabase",
                "glue:CreateDatabase"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:catalog"
            ]
        }
    ]
}
```

La seguente query di esempio elenca i database correnti.

```
SELECT * FROM SVV_EXTERNAL_DATABASES WHERE
databasename = 'example_db1' or databasename = 'example_db2';
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per elencare i database correnti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabases"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:database/example_db1",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db2",
                "arn:aws:glue:us-west-2:redshift-account:catalog"

            ]
        }
    ]
}
```

Policy per le operazioni di tabella

Affinché possano visualizzare, creare, rimuovere, modificare o effettuare altre operazioni sulle tabelle, gli utenti hanno bisogno di diversi tipi di accessi. Hanno bisogno di poter accedere alle tabelle, ai database ai quali queste appartengono e al catalogo.

La query di esempio seguente crea una tabella esterna.

```
CREATE EXTERNAL TABLE example_db.example_tbl0(
    col0 INT,
    col1 VARCHAR(255)
) PARTITIONED BY (part INT) STORED AS TEXTFILE
LOCATION 's3://test/s3/location/';
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per la creazione di una tabella esterna.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateTable"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:catalog",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/example_tbl0"
            ]
        }
    ]
}
```

Le query di esempio seguenti elencano ciascuna le tabelle esterne correnti.

```
SELECT * FROM svv_external_tables
WHERE tablename = 'example_tbl0' OR
tablename = 'example_tbl1';
```

```
SELECT * FROM svv_external_columns
WHERE tablename = 'example_tbl0' OR
tablename = 'example_tbl1';
```

```
SELECT parameters FROM svv_external_tables
WHERE tablename = 'example_tbl0' OR
tablename = 'example_tbl1';
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per elencare le tabelle esterne correnti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetTables"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:catalog",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/
example_tbl0",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/example_tbl1"

            ]
        }
    ]
}
```

La query di esempio seguente modifica una tabella esistente.

```
ALTER TABLE example_db.example_tbl0
SET TABLE PROPERTIES ('numRows' = '100');
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per la modifica di una tabella esistente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetTable",
                "glue:UpdateTable"
```

```
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:catalog",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/example_tbl0"

            ]
        }
    ]
}
```

La query di esempio seguente rimuove una tabella esistente.

```
DROP TABLE example_db.example_tbl0;
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per la rimozione di una tabella esistente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:DeleteTable"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:catalog",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/example_tbl0"
            ]
        }
    ]
}
```

Policy per le operazioni di partizione

Perché possano effettuare operazioni a livello di partizione (visualizzazione, creazione, rimozione, modifica e così via), gli utenti hanno bisogno delle autorizzazioni per le tabelle alle quali appartengono le partizioni. Hanno inoltre bisogno delle autorizzazione per i database correlati e il catalogo dati AWS Glue .

La query di esempio seguente crea una partizione.

```
ALTER TABLE example_db.example_tbl0
ADD PARTITION (part=0) LOCATION 's3://test/s3/location/part=0/';
ALTER TABLE example_db.example_t
ADD PARTITION (part=1) LOCATION 's3://test/s3/location/part=1/';
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per la creazione di una partizione.

```
 {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetTable",
                "glue:BatchCreatePartition"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:catalog",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/example_tbl0"
            ]
        }
    ]
}
```

La seguente query di esempio elenca le partizioni correnti.

```
SELECT * FROM svv_external_partitions
WHERE schemname = 'example_db' AND
tablename = 'example_tbl0'
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per elencare le partizioni correnti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetPartitions",
                "glue:GetTables",
                "glue:GetTable"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:catalog",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/example_tbl0"
            ]
        }
    ]
}
```

La query di esempio seguente modifica una partizione esistente.

```
ALTER TABLE example_db.example_tbl0 PARTITION(part='0')
SET LOCATION 's3://test/s3/new/location/part=0/';
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per la modifica di una partizione esistente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetPartition",
                "glue:UpdatePartition"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:catalog",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/example_tbl0"
            ]
        }
    ]
}
```

La query di esempio seguente rimuove una partizione esistente.

```
ALTER TABLE example_db.example_tbl0 DROP PARTITION(part='0');
```

La policy IAM seguente fornisce le autorizzazioni minime richieste per la rimozione di una partizione esistente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:DeletePartition"
            ],
            "Resource": [
                "arn:aws:glue:us-west-2:redshift-account:catalog",
                "arn:aws:glue:us-west-2:redshift-account:database/example_db",
                "arn:aws:glue:us-west-2:redshift-account:table/example_db/example_tbl0"
            ]
```

```
        }
    ]
}
```

Utilizzo di Redshift Spectrum con AWS Lake Formation

Puoi utilizzarlo AWS Lake Formation per definire e applicare centralmente policy di accesso a livello di database, tabelle e colonne ai dati archiviati in Amazon S3. Dopo aver registrato i dati con AWS Glue Data Catalog abilitato con Lake Formation, è possibile eseguire query utilizzando diversi servizi, incluso Redshift Spectrum.

Lake Formation offre la sicurezza e la governance del catalogo di dati. All'interno di Lake Formation, è possibile concedere e revocare le autorizzazioni per gli oggetti del catalogo di dati, come database, tabelle, colonne e archiviazione Amazon S3 sottostante.

> **Important**
>
> È possibile utilizzare Redshift Spectrum con un catalogo dati abilitato per Lake Formation solo AWS nelle regioni in cui è disponibile Lake Formation. Per l'elenco delle regioni disponibili, consulta Endpoint e quote di AWS Lake Formation in Riferimenti generali di AWS.

Utilizzando Redshift Spectrum con Lake Formation, è possibile effettuare le seguenti operazioni:

- Utilizzare Lake Formation come luogo centralizzato in cui concedere e revocare le autorizzazioni e accedere alle policy di controllo su tutti i dati del data lake. Lake Formation offre una gerarchia alle autorizzazioni per controllare l'accesso a database e tabelle in un catalogo di dati. Per ulteriori informazioni, consulta la pagina relativa alla panoramica delle autorizzazioni di Lake Formation nella Guida per gli sviluppatori di AWS Lake Formation .
- Crea tabelle esterne ed esegui query sui dati nel data lake. Prima che gli utenti nell'account possano eseguire le query, un amministratore dell'account data lake registra i percorsi Amazon S3 esistenti contenenti dati di origine con Lake Formation. L'amministratore crea anche tabelle e concede le autorizzazioni ai tuoi utenti. L'accesso può essere concesso per database, tabelle o colonne. L'amministratore può utilizzare i filtri di dati in Lake Formation per garantire un controllo granulare dell'accesso ai dati sensibili memorizzati in Amazon S3. Per ulteriori informazioni, consulta Utilizzo di filtri di dati per la sicurezza a livello di riga e cella.

Dopo che i dati sono stati registrati nel catalogo dati, ogni volta che gli utenti provano a eseguire le query, Lake Formation verifica l'accesso alla tabella per quel principal specifico. Lake Formation fornisce credenziali temporanee a Redshift Spectrum e la query viene eseguita.

- Esegui le query Redshift Spectrum su un dispositivo montato automaticamente AWS Glue Data Catalog utilizzando credenziali IAM ottenute con o `GetCredentials` e `GetClusterCredentials` gestisci le autorizzazioni Lake Formation per utente del database (iamr:UserName o IAM:UserName).

Quando si utilizza Redshift Spectrum con un catalogo di dati abilitato per Lake Formation, deve essere soddisfatta una delle seguenti condizioni:

- Un ruolo IAM associato al cluster con autorizzazione al catalogo dati.
- Un'identità IAM federata configurata per gestire l'accesso a risorse esterne. Per maggiori informazioni, consulta Utilizzo di un'identità federata per gestire l'accesso di Amazon Redshift alle risorse locali e alle tabelle esterne di Amazon Redshift Spectrum.

> **Important**
>
> Non è possibile collegare i ruoli IAM quando si utilizza Redshift Spectrum con un catalogo di dati abilitato per Lake Formation.

Per ulteriori informazioni sui passaggi necessari per la configurazione AWS Lake Formation per l'utilizzo con Redshift Spectrum, consulta Tutorial: Creazione di un data lake da una sorgente JDBC in Lake Formation nella Developer Guide.AWS Lake Formation In particolare, consulta Esecuzione di query sui dati nel data lake utilizzando Amazon Redshift Spectrum per i dettagli sull'integrazione con Redshift Spectrum. I dati e AWS le risorse utilizzati in questo argomento dipendono dai passaggi precedenti del tutorial.

Utilizzo di filtri di dati per la sicurezza a livello di riga e cella

Puoi definire filtri di dati AWS Lake Formation per controllare l'accesso delle query Redshift Spectrum a livello di riga e cella ai dati definiti nel tuo Data Catalog. Per configurare questo controllo, eseguirai le seguenti attività:

- Creazione di un filtro di dati in Lake Formation con le seguenti informazioni:

- Una specifica di colonna con un elenco di colonne da includere o escludere dai risultati della query.
- Un'espressione di filtro di riga che specifica le righe da includere nei risultati della query.

  Per ulteriori informazioni su come creare un filtro di dati, consulta Filtri di dati in Lake Formation nella Guida per gli sviluppatori di AWS Lake Formation .
- Creazione di una tabella esterna in Amazon Redshift che faccia riferimento a una tabella nel tuo catalogo dati abilitato per Lake Formation. Per dettagli su come eseguire query su una tabella di Lake Formation utilizzando Redshift Spectrum, consulta Esecuzione di query sui dati nel data lake utilizzando Amazon Redshift Spectrum nella Guida per gli sviluppatori di AWS Lake Formation .

Dopo aver definito la tabella in Amazon Redshift, puoi eseguire query sulla tabella di Lake Formation e accedere solo alle righe e alle colonne consentite dal filtro di dati.

Per una guida dettagliata su come configurare la sicurezza a livello di riga e di cella in Lake Formation e quindi eseguire query mediante Redshift Spectrum, consulta la pagina relativa all'utilizzo di Amazon Redshift Spectrum con policy di sicurezza a livello di riga e di cella definite in AWS Lake Formation.

Creazione di file di dati per le query in Amazon Redshift Spectrum

I file di dati utilizzati per le query in Amazon Redshift Spectrum sono in genere gli stessi tipi di file utilizzati per altre applicazioni. Ad esempio, gli stessi tipi di file vengono utilizzati con Amazon Athena, Amazon EMR e Amazon. QuickSight È possibile eseguire query sui dati nel loro formato originale direttamente da Amazon S3. Per farlo, i file di dati devono avere un formato supportato da Redshift Spectrum e trovarsi in un bucket Amazon S3 accessibile da parte del cluster.

Il bucket Amazon S3 con i file di dati e il cluster Amazon Redshift devono trovarsi nella stessa regione. AWS Per informazioni sulle AWS regioni supportate, consulta. Regioni di Amazon Redshift Spectrum

Formati di dati per Redshift Spectrum

Redshift Spectrum supporta i seguenti formati di dati strutturati e semi-strutturati:

Formato del file	Colonna	Supporta letture parallele	Unità divisa
Parquet	Sì	Sì	Gruppo di righe
ORC	Sì	Sì	Stripe
RCFile	Sì	Sì	Gruppo di righe
TextFile	No	Sì	Riga
SequenceFile	No	Sì	Riga o blocco
RegexSerde	No	Sì	Riga
OpenCSV	No	Sì	Riga
AVRO	No	Sì	Blocco
Ion	No	No	N/D
JSON	No	No	N/D

Nella tabella precedente, i titoli indicano quanto segue:

- Colonna: se il formato del file memorizza fisicamente i dati in una struttura orientata alle colonne anziché in una struttura orientata alle righe.
- Supporta letture parallele: se il formato di file supporta la lettura di singoli blocchi all'interno del file. La lettura di singoli blocchi consente l'elaborazione distribuita di un file su più richieste Redshift Spectrum indipendenti invece di dover leggere il file completo in una singola richiesta.
- Unità di divisione: per i formati di file che possono essere letti in parallelo, l'unità di divisione è il più piccolo blocco di dati che una singola richiesta Redshift Spectrum può elaborare.

> **Note**
>
> I valori di timestamp nei file di testo devono essere nel formato `yyyy-MM-dd HH:mm:ss.SSSSSS`, come il seguente valore di timestamp `2017-05-01 11:30:59.000000`.

Consigliamo di utilizzare un formato di file di storage a colonne come Apache Parquet. Con un formato di questo tipo, è possibile ridurre al minimo il trasferimento di dati al di fuori di Amazon S3 selezionando solo le colonne necessarie.

Tipi di compressione per Redshift Spectrum

Per ridurre lo spazio di archiviazione, migliorare le prestazioni e ridurre i costi, consigliamo vivamente di comprimere i file di dati. Redshift Spectrum riconosce i tipi di compressione in funzione dell'estensione dei file.

Redshift Spectrum supporta i seguenti tipi di compressione ed estensioni:

Algoritmo di compressione	Estensione di file	Supporta letture parallele
Gzip	.gz	No
Bzip2	.bz2	Sì
Snappy	.snappy	No

Puoi applicare la compressione a diversi livelli. Più comunemente, comprimi un intero file o comprimi singoli blocchi all'interno di un file. La compressione dei formati colonnari a livello di file non comporta vantaggi in termini di prestazioni.

Affinché Redshift Spectrum possa leggere un file in parallelo, deve essere vero quanto segue:

- Il formato di file supporta le letture parallele.
- La compressione a livello di file, se presente, supporta le letture parallele.

Non importa se le singole unità di divisione all'interno di un file vengono compresse utilizzando un algoritmo di compressione che può essere letto in parallelo, perché ogni unità divise viene elaborata da una singola richiesta Redshift Spectrum. Un esempio di questo sono i file Parquet compressi con Snappy. I singoli gruppi di righe all'interno del file Parquet vengono compressi utilizzando Snappy, ma la struttura di livello superiore del file rimane decompressa. In questo caso, il file può essere letto in parallelo perché ogni richiesta Redshift Spectrum può leggere ed elaborare singoli gruppi di righe da Amazon S3.

Crittografia per Redshift Spectrum

Redshift Spectrum decrittografa in modo trasparente i file di dati crittografati utilizzando le seguenti opzioni di crittografia:

- Crittografia lato server (SSE-S3) con una chiave di crittografia AES-256 gestita da Amazon S3.
- Crittografia lato server con chiavi gestite da AWS Key Management Service (SSE-KMS).

Redshift Spectrum non supporta la crittografia lato client di Amazon S3. Per ulteriori informazioni sulla crittografia lato server, consultare Protezione dei dati con la crittografia lato server nella Guida per l'utente di Amazon Simple Storage Service.

Amazon Redshift utilizza l'elaborazione MPP (Massively Parallel Processing) per eseguire rapidamente query complesse su grandi quantità di dati. Redshift Spectrum applica lo stesso principio per eseguire le query di dati esterni, utilizzando molteplici istanze di Redshift Spectrum per eseguire la scansione di file. Inserisci i file in una cartella distinta per ogni tabella.

Puoi ottimizzare i dati per l'elaborazione parallela procedendo come segue:

- Se il formato di file o la compressione non supporta la lettura in parallelo, fraziona file di grandi dimensioni in molti file più piccoli. Ti consigliamo di utilizzare file di dimensioni comprese tra 64 MB e 1 GB.
- Fai in modo che tutti i file siano all'incirca della stessa dimensione. Se alcuni file sono più grandi di altri, Redshift Spectrum non può distribuire il carico di lavoro in modo uniforme.

Creazione di schemi esterni per Amazon Redshift Spectrum

Tutte le tabelle esterne devono essere create in uno schema esterno, che crei utilizzando un'istruzione CREATE EXTERNAL SCHEMA.

> **Note**
>
> Alcune applicazioni utilizzano i termini database e schema in modo interscambiabile. In Amazon Redshift viene utilizzato il termine schema.

Uno schema esterno Amazon Redshift fa riferimento a un database esterno in un catalogo dati esterno. È possibile creare un database esterno in Amazon Redshift, in Amazon Athena, in AWS

Glue Data Catalog o in un metastore Apache Hive, come Amazon EMR. Se si crea un database esterno in Amazon Redshift, il database si trova nel catalogo dati di Athena. Per creare un database in un metastore Hive, devi crearlo nell'applicazione Hive.

Amazon Redshift deve essere autorizzato ad accedere al catalogo di dati in Athena e ai file di dati in Amazon S3 per tuo conto. Per fornire tale autorizzazione, devi prima creare un AWS Identity and Access Management ruolo (IAM). Successivamente, si collega il ruolo al cluster e si fornisce l'Amazon Resource Name (ARN) per il ruolo nell'istruzione `CREATE EXTERNAL SCHEMA` di Amazon Redshift. Per ulteriori informazioni sull'autorizzazione , consultare Policy IAM per Amazon Redshift Spectrum.

> **Note**
>
> Se attualmente disponi di tabelle esterne Redshift Spectrum nell'Athena Data Catalog, puoi migrare il tuo Athena Data Catalog in un Data Catalog. AWS Glue Per utilizzare un catalogo AWS Glue dati con Redshift Spectrum, potrebbe essere necessario modificare le policy IAM. Per ulteriori informazioni, consulta Upgrade to the AWS Glue Data Catalog nella Amazon Athena User Guide.

Per creare un database esterno nello stesso tempo, devi creare uno schema esterno, specificare `FROM DATA CATALOG` e includere la clausola `CREATE EXTERNAL DATABASE` nell'istruzione `CREATE EXTERNAL SCHEMA`.

L'esempio seguente crea uno schema esterno denominato `spectrum_schema` utilizzando il database esterno `spectrum_db`.

```
create external schema spectrum_schema from data catalog
database 'spectrum_db'
iam_role 'arn:aws:iam::123456789012:role/MySpectrumRole'
create external database if not exists;
```

Se il catalogo dati viene gestito mediante Athena, specificare il nome di database Athena e la regione AWS in cui si trova il catalogo dati Athena.

Nell'esempio seguente viene creato uno schema esterno che utilizza il database `sampledb` di default nel catalogo dati di Athena.

```
create external schema athena_schema from data catalog
database 'sampledb'
```

```
iam_role 'arn:aws:iam::123456789012:role/MySpectrumRole'
region 'us-east-2';
```

> **Note**
>
> Il `region` parametro fa riferimento alla AWS regione in cui si trova il catalogo dati Athena, non alla posizione dei file di dati in Amazon S3.

Se il catalogo di dati viene gestito mediante un metastore Hive, come Amazon EMR, i gruppi di sicurezza devono essere configurati per consentire il traffico tra i cluster.

Nell'istruzione CREATE EXTERNAL SCHEMA, specifica `FROM HIVE METASTORE` e includi l'URI e il numero di porta del metastore. L'esempio seguente crea uno schema esterno che utilizza un database metastore Hive denominato `hive_db`.

```
create external schema hive_schema
from hive metastore
database 'hive_db'
uri '172.10.10.10' port 99
iam_role 'arn:aws:iam::123456789012:role/MySpectrumRole'
```

Per visualizzare gli schemi esterni per il cluster, eseguire una query sulla tabella di catalogo PG_EXTERNAL_SCHEMA o sulla vista SVV_EXTERNAL_SCHEMAS. L'esempio seguente esegue una query su SVV_EXTERNAL_SCHEMAS, che unisce in join PG_EXTERNAL_SCHEMA e PG_NAMESPACE.

```
select * from svv_external_schemas
```

Per la sintassi completa del comando e alcuni esempi, consultare CREATE EXTERNAL SCHEMA.

Utilizzo di cataloghi esterni in Amazon Redshift Spectrum

I metadati per i database esterni e le tabelle esterne di Amazon Redshift Spectrum sono archiviati in un catalogo dati esterno. Per impostazione predefinita, i metadati di Redshift Spectrum sono archiviati nel catalogo dati di Athena. È possibile visualizzare e gestire i database e le tabelle di Redshift Spectrum nella console Athena.

È inoltre possibile creare e gestire i database esterni e le tabelle esterne utilizzando il linguaggio DDL (Data Definition Language) Hive con Athena o un metastore Hive come Amazon EMR.

> **Note**
>
> Consigliamo di utilizzare Amazon Redshift per creare e gestire i database esterni e le tabelle esterne in Redshift Spectrum.

Visualizzazione dei database Redshift Spectrum in Athena e AWS Glue

È possibile creare un database esterno includendo la clausola CREATE EXTERNAL DATABASE IF NOT EXISTS nell'istruzione CREATE EXTERNAL SCHEMA. In questi casi, i metadati del database esterno sono archiviati nel catalogo dati. I metadati per le tabelle esterne che crei qualificati dallo schema esterno sono archiviati anche nel tuo catalogo dati .

Athena e AWS Glue mantieni un catalogo dati per ogni supporto. Regione AWS Per visualizzare i metadati della tabella, accedi ad Athena AWS Glue o alla console. In Athena, scegli Origini dati, le tue AWS Glue, quindi visualizza i dettagli del tuo database. In AWS Glue, scegli Database, il tuo database esterno, quindi visualizza i dettagli del tuo database.

Se le tabelle esterne vengono create e gestite mediante Athena, registrare il database utilizzando CREATE EXTERNAL SCHEMA. Ad esempio, il comando seguente registra il database Athena denominato `sampledb`.

```
create external schema athena_sample
from data catalog
database 'sampledb'
iam_role 'arn:aws:iam::123456789012:role/mySpectrumRole'
region 'us-east-1';
```

Quando si esegue una query sulla vista di sistema SVV_EXTERNAL_TABLES, vengono visualizzate le le tabelle nel database `sampledb` Athena e le tabelle create in Amazon Redshift.

```
select * from svv_external_tables;
```

```
schemaname    | tablename        | location

--------------+------------------
+-----------------------------------------------------
athena_sample | elb_logs         | s3://athena-examples/elb/plaintext
athena_sample | lineitem_1t_csv  | s3://myspectrum/tpch/1000/lineitem_csv
```

```
athena_sample | lineitem_1t_part | s3://myspectrum/tpch/1000/lineitem_partition

spectrum      | sales            | s3://redshift-downloads/tickit/spectrum/sales

spectrum      | sales_part       | s3://redshift-downloads/tickit/spectrum/sales_part
```

Registrazione di un database di un metastore Apache Hive

Se crei delle tabelle esterne in un metastore Apache Hive, puoi utilizzare CREATE EXTERNAL SCHEMA per registrare tali tabelle in Redshift Spectrum.

Nell'istruzione CREATE EXTERNAL SCHEMA, specifica la clausola FROM HIVE METASTORE e fornisci l'URI e il numero di porta del metastore Hive. Il ruolo IAM deve includere l'autorizzazione per accedere ad Amazon S3 ma nessuna autorizzazione Athena. L'esempio seguente registra un metastore Hive.

```
create external schema if not exists hive_schema
from hive metastore
database 'hive_database'
uri 'ip-10-0-111-111.us-west-2.compute.internal' port 9083
iam_role 'arn:aws:iam::123456789012:role/mySpectrumRole';
```

Abilitazione del cluster Amazon Redshift per accedere al cluster Amazon EMR

Se il metastore Hive è in Amazon EMR, è necessario fornire al cluster Amazon Redshift l'accesso al cluster Amazon EMR. Per farlo, è necessario creare un gruppo di sicurezza Amazon EC2. Quindi si consente tutto il traffico in entrata verso il gruppo di sicurezza EC2 dal gruppo di sicurezza del cluster Amazon Redshift e dal gruppo di sicurezza del cluster Amazon EMR. Quindi viene aggiunta la sicurezza EC2 al cluster Amazon Redshift e al cluster Amazon EMR.

Visualizzare il nome del gruppo di sicurezza del cluster Amazon Redshift

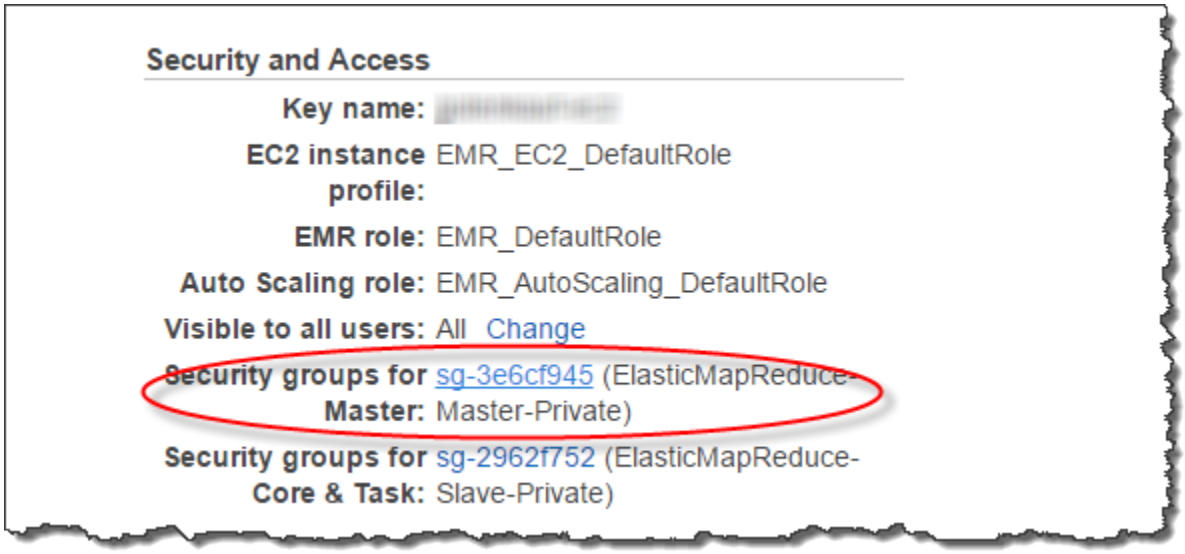
Per visualizzare il gruppo di sicurezza, procedere come segue:

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione scegliere Clusters (Cluster), quindi scegliere dall'elenco il cluster per visualizzarne i dettagli.
3. Scegliere Properties (Proprietà) e visualizzare la sezione Network and security settings (Impostazioni rete e sicurezza).

4. Trova il gruppo di sicurezza in Gruppo di sicurezza VPC e prendi nota.

Visualizzare il nome del gruppo di sicurezza dei nodi master di Amazon EMR

1. Aprire il cluster Amazon EMR. Per ulteriori informazioni, consulta Utilizzo delle configurazioni di sicurezza per impostare la sicurezza del cluster nella Guida alla gestione di Amazon EMR.
2. In Sicurezza e accesso, prendere nota del nome del gruppo di sicurezza del nodo principale Amazon EMR.



Per creare o modificare un gruppo di sicurezza Amazon EC2 per consentire la connessione tra Amazon Redshift ed Amazon EMR

1. Nel pannello di controllo di Amazon EC2, scegliere Gruppi di sicurezza. Per ulteriori informazioni, consulta le regole dei gruppi di sicurezza nella Guida per l'utente di Amazon EC2
2. Scegliere Create Security Group (Crea gruppo di sicurezza).
3. Se si sta utilizzando VPC, scegliere quello in cui si trovano i cluster Amazon Redshift ed Amazon EMR.
4. Aggiungere una regola in entrata.
    1. Per Type (Tipo), scegliere Custom TCP (TCP personalizzato).
    2. In Source (Origine), scegliere Custom (Personalizzata).
    3. Digitare il nome del gruppo di sicurezza Amazon Redshift.
5. Aggiungere un'altra regola in entrata.

1. Per Type (Tipo), scegliere TCP.
2. Alla voce Port Range (Intervallo porte), inserire 9083.

> **Note**
>
> La porta predefinita per un HMS EMR è 9083. Se HMS utilizza una porta differente, specificare la porta nella regola in entrata e nella definizione dello schema esterno.

3. In Source (Origine), scegliere Custom (Personalizzata).

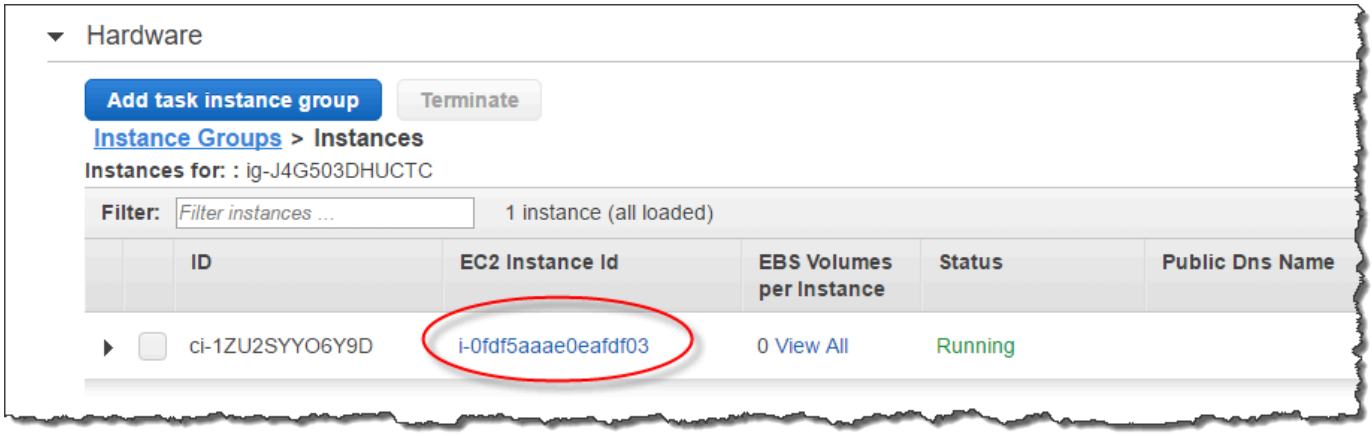
6. Inserire un nome e una descrizione del gruppo di sicurezza.
7. Scegliere Create Security Group (Crea gruppo di sicurezza).

Per aggiungere il gruppo di sicurezza Amazon EC2 creato nella procedura precedente al cluster Amazon Redshift

1. Nella console Amazon Redshift, scegliere il cluster.
2. Scegli Properties (Proprietà).
3. Visualizzare Impostazioni di rete e sicurezza e scegliere Modificare.
4. Nello stato Gruppo di sicurezza VPC, scegliere il nome del nuovo gruppo di sicurezza.
5. Seleziona Salvataggio delle modifiche.

Per aggiungere il gruppo di sicurezza Amazon EC2 al cluster Amazon EMR

1. In Amazon EMR, scegliere il cluster. Per ulteriori informazioni, consulta Utilizzo configurazioni sicurezza per impostare la sicurezza del cluster nella Guida alla gestione di Amazon EMR.
2. In Hardware, scegliere il collegamento per il nodo master.
3. Scegliere il collegamento nella colonna EC2 instance ID (ID istanza EC2).



4. Per Azioni, scegliere Sicurezza, Modifica gruppi di sicurezza.
5. Nello statoGruppi di sicurezza associati, scegliere il nuovo gruppo di sicurezza e scegliere Aggiungi gruppo di sicurezza.
6. Selezionare Salva.

Creazione di tabelle esterne per Redshift Spectrum

Crei una tabella esterna in uno schema esterno. Per creare tabelle esterne, devi essere il proprietario dello schema esterno o un utente con privilegi avanzati. Per trasferire la proprietà di uno schema esterno, utilizza ALTER SCHEMA per cambiare il proprietario. L'esempio seguente cambia il proprietario dello schema `spectrum_schema` in `newowner`.

```
alter schema spectrum_schema owner to newowner;
```

Per eseguire una query di Redshift Spectrum, sono necessarie le seguenti autorizzazioni:

- Autorizzazione di utilizzare lo schema
- Autorizzazione di creare tabelle temporanee nel database corrente

L'esempio seguente concede l'autorizzazione all'utilizzo dello schema `spectrum_schema` al gruppo di utenti `spectrumusers`.

```
grant usage on schema spectrum_schema to group spectrumusers;
```

L'esempio seguente concede l'autorizzazione temporanea per il database `spectrumdb` al gruppo di utenti `spectrumusers`.

```
grant temp on database spectrumdb to group spectrumusers;
```

Puoi creare una tabella esterna in Amazon Redshift AWS Glue, Amazon Athena o un metastore Apache Hive. Per ulteriori informazioni, consultare Nozioni di base sull'uso di AWS Glue nella Guida per gli sviluppatori di AWS Glue , Nozioni di base nella Guida per l'utente di Amazon Athena oppure Apache Hive nella Guida per gli sviluppatori di Amazon EMR.

Se la tua tabella esterna è definita in AWS Glue, Athena o in un metastore Hive, devi prima creare uno schema esterno che faccia riferimento al database esterno. È quindi possibile fare riferimento alla tabella esterna nell'istruzione SELECT aggiungendo un prefisso al nome della tabella con il nome dello schema senza che sia necessario creare la tabella in Amazon Redshift. Per ulteriori informazioni, consulta Creazione di schemi esterni per Amazon Redshift Spectrum.

Per consentire ad Amazon Redshift di visualizzare le tabelle in AWS Glue Data Catalog, aggiungi il ruolo `glue:GetTable` IAM di Amazon Redshift. In caso contrario, si potrebbe verificare un errore simile al seguente:

```
RedshiftIamRoleSession is not authorized to perform: glue:GetTable on resource: *;
```

Ad esempio, si supponga di avere una tabella esterna denominata `lineitem_athena` definita nel catalogo esterno di Athena. In tal caso, puoi definire uno schema esterno denominato `athena_schema` e quindi eseguire una query sulla tabella utilizzando l'istruzione SELECT seguente.

```
select count(*) from athena_schema.lineitem_athena;
```

Per definire una tabella esterna in Amazon Redshift, utilizzare il comando CREATE EXTERNAL TABLE. L'istruzione della tabella esterna definisce le colonne della tabella, il formato dei file di dati e la posizione dei dati in Amazon S3. Redshift Spectrum esegue la scansione dei file nella cartella specificata e in tutte le sottocartelle. Redshift Spectrum ignora i file nascosti e i file che iniziano con un punto, un carattere di sottolineatura o un marcatore hash (. , _ o #) oppure che terminano con una tilde (~).

Nell'esempio seguente viene creata una tabella denominata SALES nello schema esterno Amazon Redshift denominato `spectrum`. I dati sono in file di testo delimitati da tabulazioni.

```
create external table spectrum.sales(
salesid integer,
```

```
listid integer,
sellerid integer,
buyerid integer,
eventid integer,
dateid smallint,
qtysold smallint,
pricepaid decimal(8,2),
commission decimal(8,2),
saletime timestamp)
row format delimited
fields terminated by '\t'
stored as textfile
location 's3://redshift-downloads/tickit/spectrum/sales/'
table properties ('numRows'='172000');
```

Per visualizzare le tabelle esterne, eseguire una query sulla vista di sistema SVV_EXTERNAL_TABLES.

Pseudocolonne

Di default, Amazon Redshift crea tabelle esterne con le pseudocolonne `$path`, `$size` e `$spectrum_oid`. Selezionare la colonna `$path` per visualizzare il percorso ai file di dati su Simple Storage Service (Amazon S3) e selezionare la colonna `$size` per visualizzare le dimensioni dei file di dati per ogni riga restituita da una query. La colonna `$spectrum_oid` consente di eseguire query correlate con Redshift Spectrum. Per vedere un esempio, consulta Esempio: esecuzione di sottoquery correlate in Redshift Spectrum. I nomi di colonna `$path`, `$size` e `$spectrum_oid` devono essere delimitati da virgolette doppie. Una clausola SELECT * non restituisce le pseudocolonne. È necessario includere in modo esplicito i nomi delle colonne `$path`, `$size` e `$spectrum_oid` nella query, come indicato nel seguente esempio.

```
select "$path", "$size", "$spectrum_oid"
from spectrum.sales_part where saledate = '2008-12-01';
```

Puoi disabilitare la creazione di pseudocolonne per una sessione impostando il parametro di configurazione `spectrum_enable_pseudo_columns` su `false`. Per ulteriori informazioni, consulta spectrum_enable_pseudo_columns. È anche possibile disabilitare solo la pseudocolonna `$spectrum_oid` impostando il parametro di configurazione `enable_spectrum_oid` su `false`. Per ulteriori informazioni, consulta enable_spectrum_oid. Tuttavia, disabilitando la pseudocolonna `$spectrum_oid` viene disabilitato anche il supporto per le query correlate con Redshift Spectrum.

> **Important**
>
> La selezione di `$size`, `$path` o `$spectrum_oid` comporta dei costi perché Redshift Spectrum analizza i file di dati su Simple Storage Service (Amazon S3) per determinare la dimensione del set di risultati. Per ulteriori informazioni sui prezzi, consultare Prezzi di Amazon Redshift.

Esempio di pseudocolonne

L'esempio seguente restituisce la dimensione totale dei file di dati correlati per una tabella esterna.

```
select distinct "$path", "$size"
from spectrum.sales_part;

 $path                                                                      | $size
----------------------------------------------------------------------------+-------
s3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-01/ |  1616
s3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-02/ |  1444
s3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-03/ |  1644
```

Partizionamento delle tabelle esterne di Redshift Spectrum

Quando esegui la partizione dei dati, puoi limitare la quantità di dati sottoposti a scansione da Redshift Spectrum filtrandoli in base a qualsiasi chiave di partizione.

In genere, si partizionano i dati in base a criteri temporali. Ad esempio, puoi scegliere di eseguire il partizionamento per anno, mese, data e ora. Se hai dati da più origini, puoi eseguire il partizionamento in base a una data e a un identificatore di origine dati.

La procedura seguente descrive come partizionare i dati.

Per partizionare i dati

1. Archiviare i dati in cartelle di Amazon S3 in funzione della chiave di partizione.

   Creare una cartella per ogni valore di partizione e assegnare un nome alla cartella con la chiave e il valore di partizione. Ad esempio, se si esegue la partizione per data, le cartelle possono essere denominate `saledate=2017-04-01`, `saledate=2017-04-02` e così via. Redshift Spectrum esegue la scansione dei file nella cartella di partizione e in tutte le

sottocartelle. Redshift Spectrum ignora i file nascosti e i file che iniziano con un punto, un carattere di sottolineatura o un marcatore hash (. , _ o #) oppure che terminano con una tilde (~).

2. Creare una tabella esterna e specificare la chiave di partizione nella clausola PARTITIONED BY.

   La chiave di partizione non può essere Il nome di una colonna della tabella. Il tipo di dati può essere SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE o TIMESTAMP.

3. Aggiungere le partizioni.

   Utilizzando ALTER TABLE … ADD PARTITION, aggiungere ogni partizione, specificando la colonna della partizione e il valore della chiave nonché la posizione della cartella di partizione in Amazon S3. È possibile aggiungere più partizioni in una singola istruzione ALTER TABLE … ADD. L'esempio seguente aggiunge partizioni per `'2008-01'` e `'2008-03'`.

```
alter table spectrum.sales_part add
partition(saledate='2008-01-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/
saledate=2008-01/'
partition(saledate='2008-03-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/
saledate=2008-03/';
```

> **Note**
>
> Se utilizzi il AWS Glue catalogo, puoi aggiungere fino a 100 partizioni utilizzando una singola istruzione ALTER TABLE.

Partizionamento di esempi di dati

In questo esempio, crei una tabella esterna partizionata con una singola chiave di partizione e una tabella esterna partizionata con due chiavi di partizione.

I dati di esempio per questo esempio si trovano in un bucket Amazon S3 che consente l'accesso in lettura a tutti gli utenti autenticati. AWS Il cluster e i file di dati esterni devono trovarsi nella stessa Regione AWS. Il bucket di dati di esempio si trova nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1). Per accedere ai dati mediante Redshift Spectrum, anche il cluster deve essere nella regione us-east-1. Per elencare le cartelle in Amazon S3, emettere il comando seguente.

```
aws s3 ls s3://redshift-downloads/tickit/spectrum/sales_partition/
```

```
PRE saledate=2008-01/
PRE saledate=2008-03/
PRE saledate=2008-04/
PRE saledate=2008-05/
PRE saledate=2008-06/
PRE saledate=2008-12/
```

Se non hai ancora uno schema esterno, esegui il comando seguente. Sostituisci il AWS Identity and Access Management tuo ruolo (IAM) con Amazon Resource Name (ARN).

```
create external schema spectrum
from data catalog
database 'spectrumdb'
iam_role 'arn:aws:iam::123456789012:role/myspectrumrole'
create external database if not exists;
```

Esempio 1: partizionamento con una singola chiave di partizione

Nell'esempio seguente, crei una tabella esterna partizionata per mese.

Per creare una tabella esterna partizionata per mese, esegui il comando seguente.

```
create external table spectrum.sales_part(
salesid integer,
listid integer,
sellerid integer,
buyerid integer,
eventid integer,
dateid smallint,
qtysold smallint,
pricepaid decimal(8,2),
commission decimal(8,2),
saletime timestamp)
partitioned by (saledate char(10))
row format delimited
fields terminated by '|'
stored as textfile
location 's3://redshift-downloads/tickit/spectrum/sales_partition/'
table properties ('numRows'='172000');
```

Per aggiungere le partizioni, esegui il comando ALTER TABLE seguente.

```
alter table spectrum.sales_part add
partition(saledate='2008-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-01/'

partition(saledate='2008-03')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-03/'

partition(saledate='2008-04')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-04/';
```

Per selezionare i dati dalla tabella partizionata, esegui la seguente query.

```
select top 5 spectrum.sales_part.eventid, sum(spectrum.sales_part.pricepaid)
from spectrum.sales_part, event
where spectrum.sales_part.eventid = event.eventid
  and spectrum.sales_part.pricepaid > 30
  and saledate = '2008-01'
group by spectrum.sales_part.eventid
order by 2 desc;
```

```
eventid | sum
--------+---------
   4124 | 21179.00
   1924 | 20569.00
   2294 | 18830.00
   2260 | 17669.00
   6032 | 17265.00
```

Per visualizzare le partizioni delle tabelle, eseguire una query sulla vista di sistema SVV_EXTERNAL_PARTITIONS.

```
select schemaname, tablename, values, location from svv_external_partitions
where tablename = 'sales_part';
```

```
schemaname | tablename  | values      | location

-----------+------------+-------------
+-------------------------------------------------------------------------
```

```
spectrum   | sales_part | ["2008-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-01
spectrum   | sales_part | ["2008-03"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-03
spectrum   | sales_part | ["2008-04"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-04
```

Esempio 2: partizionamento con più chiavi di partizione

Per creare una tabella esterna partizionata per `date` e `eventid`, esegui il comando seguente.

```
create external table spectrum.sales_event(
salesid integer,
listid integer,
sellerid integer,
buyerid integer,
eventid integer,
dateid smallint,
qtysold smallint,
pricepaid decimal(8,2),
commission decimal(8,2),
saletime timestamp)
partitioned by (salesmonth char(10), event integer)
row format delimited
fields terminated by '|'
stored as textfile
location 's3://redshift-downloads/tickit/spectrum/salesevent/'
table properties ('numRows'='172000');
```

Per aggiungere le partizioni, esegui il comando ALTER TABLE seguente.

```
alter table spectrum.sales_event add
partition(salesmonth='2008-01', event='101')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-01/
event=101/'

partition(salesmonth='2008-01', event='102')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-01/
event=102/'

partition(salesmonth='2008-01', event='103')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-01/
event=103/'
```

```
partition(salesmonth='2008-02', event='101')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-02/
event=101/'

partition(salesmonth='2008-02', event='102')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-02/
event=102/'

partition(salesmonth='2008-02', event='103')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-02/
event=103/'

partition(salesmonth='2008-03', event='101')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-03/
event=101/'

partition(salesmonth='2008-03', event='102')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-03/
event=102/'

partition(salesmonth='2008-03', event='103')
location 's3://redshift-downloads/tickit/spectrum/salesevent/salesmonth=2008-03/
event=103/';
```

Esegui la query seguente per selezionare i dati dalla tabella partizionata.

```
select spectrum.sales_event.salesmonth, event.eventname,
 sum(spectrum.sales_event.pricepaid)
from spectrum.sales_event, event
where spectrum.sales_event.eventid = event.eventid
  and salesmonth = '2008-02'
 and (event = '101'
 or event = '102'
 or event = '103')
group by event.eventname, spectrum.sales_event.salesmonth
order by 3 desc;
```

```
salesmonth | eventname       | sum
-----------+-----------------+--------
2008-02    | The Magic Flute | 5062.00
2008-02    | La Sonnambula   | 3498.00
```

```
2008-02    | Die Walkure     |  534.00
```

Mappatura delle colonne di una tabella esterna alle colonne ORC

Utilizzare le tabelle esterne di Amazon Redshift Spectrum per eseguire query sui dati dai file nel formato ORC. Il formato Optimized Row Columnar (ORC) è un formato di file di storage a colonne che supporta le strutture di dati annidate. Per ulteriori informazioni sull'esecuzione di query su dati nidificati, consultare Esecuzione di query su dati nidificati con Amazon Redshift Spectrum.

Quando crei una tabella esterna che fa riferimento ai dati in un file ORC, devi mappare ogni colonna della tabella verso una colonna nei dati ORC. A tale scopo, utilizza uno dei seguenti metodi:

- Mappatura in base alla posizione
- Mappatura in base al nome delle colonne

La mappatura in base al nome delle colonne è l'opzione predefinita.

Mappatura in base alla posizione

Con la mappatura in base alla posizione, la prima colonna definita nella tabella esterna mappa alla prima colonna nel file di dati ORC, la seconda alla seconda e così via. Per la mappatura in base alla posizione, è necessario che l'ordine delle colonne nella tabella esterna e nel file ORC corrisponda. Diversamente, puoi mappare le colonne in base al nome.

> **Important**
>
> Nelle release precedenti, Redshift Spectrum utilizzava la mappatura in base alla posizione come opzione predefinita. Se devi continuare a utilizzare la mappatura in base alla posizione per le tabelle esistenti, imposta la proprietà di tabella `orc.schema.resolution` su `position`, come mostrato nell'esempio seguente.
>
> ```
> alter table spectrum.orc_example
> set table properties('orc.schema.resolution'='position');
> ```

Ad esempio, la tabella `SPECTRUM.ORC_EXAMPLE` è definita come segue.

```
create external table spectrum.orc_example(
```

```
int_col int,
float_col float,
nested_col struct<
  "int_col" : int,
  "map_col" : map<int, array<float >>
   >
) stored as orc
location 's3://example/orc/files/';
```

La struttura della tabella può essere astratta come segue.

```
• 'int_col' : int
• 'float_col' : float
• 'nested_col' : struct
   o 'int_col' : int
   o 'map_col' : map
      - key : int
      - value : array
         - value : float
```

Il file ORC sottostante presenta la seguente struttura di file.

```
• ORC file root(id = 0)
   o 'int_col' : int (id = 1)
   o 'float_col' : float (id = 2)
   o 'nested_col' : struct (id = 3)
      - 'int_col' : int (id = 4)
      - 'map_col' : map (id = 5)
         - key : int (id = 6)
         - value : array (id = 7)
            - value : float (id = 8)
```

In questo esempio, puoi mappare ciascuna colonna nella tabella esterna verso una colonna nel file di dati ORC rigorosamente in base alla posizione. Di seguito è riportata la mappatura.

Nome della colonna della tabella esterna	ID della colonna ORC	Nome della colonna ORC
int_col	1	int_col
float_col	2	float_col

Nome della colonna della tabella esterna	ID della colonna ORC	Nome della colonna ORC
nested_col	3	nested_col
nested_col.int_col	4	int_col
nested_col.map_col	5	map_col
nested_col.map_col.key	6	ND
nested_col.map_col.value	7	ND
nested_col.map_col.value.item	8	N/A

Mappatura in base al nome delle colonne

Tramite la mappatura dei nomi, puoi mappare le colonne in una tabella esterna verso colonne con nome nei file ORC allo stesso livello e con il medesimo nome.

Supponi ad esempio di voler mappare la tabella del precedente esempio, `SPECTRUM.ORC_EXAMPLE`, con un file ORC che utilizza la seguente struttura di file.

```
• ORC file root(id = 0)
   o 'nested_col' : struct (id = 1)
      - 'map_col' : map (id = 2)
         - key : int (id = 3)
         - value : array (id = 4)
            - value : float (id = 5)
      - 'int_col' : int (id = 6)
   o 'int_col' : int (id = 7)
   o 'float_col' : float (id = 8)
```

Utilizzando la mappatura basata sulla posizione, Redshift Spectrum tenta di eseguire la mappatura seguente.

Nome della colonna della tabella esterna	ID della colonna ORC	Nome della colonna ORC
int_col	1	struct
float_col	7	int_col
nested_col	8	float_col

Quando si esegue una query su una tabella con la precedente mappatura basata sulla posizione, il comando SELECT ha esito negativo sulla convalida del tipo in quanto le strutture sono diverse.

Puoi mappare la stessa tabella esterna verso entrambe le strutture di file mostrate negli esempi precedenti utilizzando la mappatura basata sui nomi. Le colonne di tabella `int_col`, `float_col` e `nested_col` vengono mappate in base al nome verso le colonne con i medesimi nomi nel file ORC. La colonna denominata `nested_col` nella tabella esterna è una colonna `struct` con sottocolonne denominate `map_col` e `int_col`. Le sottocolonne vengono inoltre mappate correttamente alle colonne corrispondenti nel file ORC in base al nome.

Creazione di tabelle esterne per i dati gestiti in Apache Hudi

Per eseguire query sui dati in formato Apache Hudi Copy On Write (CoW), è possibile utilizzare le tabelle esterne di Amazon Redshift Spectrum. Una tabella Hudi Copy On Write è una raccolta di file Apache Parquet archiviati in Amazon S3. È possibile leggere le tabelle Copy On Write (CoW) nelle versioni 0.5.2, 0.6.0, 0.7.0, 0.8.0, 0.9.0, 0.10.0, 0.10.1, 0.11.0 e 0.11.1 di Apache Hudi create e modificate mediante operazioni di scrittura insert, delete e upsert. Le tabelle bootstrap, ad esempio, non sono supportate. Per ulteriori informazioni, consultare Copia sulla tabella di scrittura nella documentazione open source di Apache Hudi.

Quando si crea una tabella esterna che fa riferimento ai dati in formato Hudi CoW, è necessario mappare ogni colonna della tabella esterna a una colonna nei dati Hudi. La mappatura viene eseguita per colonna.

Le istruzioni DDL (Data Definition Language) per tabelle Hudi partizionate e non partizionate sono simili a quelle degli altri formati di file Apache Parquet. Per le tabelle Hudi, `INPUTFORMAT` può essere definito come `org.apache.hudi.hadoop.HoodieParquetInputFormat`. Il parametro `LOCATION` deve puntare alla cartella di base della tabella Hudi che contiene la cartella `.hoodie`,

necessaria per stabilire la tempistica di commit Hudi. In alcuni casi, un'operazione SELECT su una tabella Hudi potrebbe non riuscire con il messaggio Nessuna tempistica di commit Hudi valida trovata. In questo caso, verificare se la cartella `.hoodie` si trova nella posizione corretta e se contiene una tempistica di commit Hudi valida.

> **Note**
>
> Il formato Apache Hudi è supportato solo quando si utilizza un AWS Glue Data Catalog. Non è supportato se si utilizza un metastore Apache Hive come catalogo esterno.

Il formato DDL per definire una tabella non partizionata è il seguente.

```
CREATE EXTERNAL TABLE tbl_name (columns)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS
INPUTFORMAT 'org.apache.hudi.hadoop.HoodieParquetInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION 's3://s3-bucket/prefix'
```

Il formato DDL per definire una tabella partizionata è il seguente.

```
CREATE EXTERNAL TABLE tbl_name (columns)
PARTITIONED BY(pcolumn1 pcolumn1-type[,...])
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS
INPUTFORMAT 'org.apache.hudi.hadoop.HoodieParquetInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION 's3://s3-bucket/prefix'
```

Per aggiungere partizioni a una tabella Hudi partizionata, eseguire un comando ALTER TABLE ADD PARTITION in cui il parametro `LOCATION` punta alla sottocartella Amazon S3 con i file che appartengono alla partizione.

Il formato DDL per aggiungere le partizioni è il seguente.

```
ALTER TABLE tbl_name
ADD IF NOT EXISTS PARTITION(pcolumn1=pvalue1[,...])
LOCATION 's3://s3-bucket/prefix/partition-path'
```

Creazione di tabelle esterne per i dati gestiti in Delta Lake

Per eseguire una query sui dati nelle tabelle Delta Lake, è possibile utilizzare le tabelle esterne di Amazon Redshift Spectrum.

Per accedere a una tabella Delta Lake da Redshift Spectrum, generare un manifest prima della query. Un manifest Delta Lake contiene un elenco di file che costituiscono uno snapshot coerente della tabella Delta Lake. In una tabella partizionata, esiste un solo manifest per partizione. Una tabella Delta Lake è una raccolta di file Apache Parquet archiviati in Amazon S3. Per ulteriori informazioni, consultare Delta Lake nella documentazione open source di Delta Lake.

Quando si crea una tabella esterna che fa riferimento ai dati nelle tabelle Delta Lake, viene mappata ogni colonna della tabella a una colonna nella tabella Delta Lake. La mappatura viene eseguita per nome della colonna.

La DDL per le tabelle Delta Lake partizionate e non partizionate è simile a quella degli altri formati di file Apache Parquet. Per le tabelle Delta Lake, è possibile definire `INPUTFORMAT` come `org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat` e `OUTPUTFORMAT` come `org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat`. Il parametro `LOCATION` deve puntare alla cartella del manifest nella cartella di base della tabella. Se un'operazione SELECT su una tabella Delta Lake non riesce per vari possibili motivi, consultare Limitazioni e risoluzione dei problemi per le tabelle Delta Lake.

Il formato DDL per definire una tabella non partizionata è il seguente.

```
CREATE EXTERNAL TABLE tbl_name (columns)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS
INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://s3-bucket/prefix/_symlink_format_manifest'
```

Il formato DDL per definire una tabella partizionata è il seguente.

```
CREATE EXTERNAL TABLE tbl_name (columns)
PARTITIONED BY(pcolumn1 pcolumn1-type[,...])
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS
INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
```

```
LOCATION 's3://s3-bucket>/prefix/_symlink_format_manifest'
```

Per aggiungere partizioni a una tabella Delta Lake partizionata, eseguire un comando ALTER TABLE ADD PARTITION in cui il parametro LOCATION punta alla sottocartella Amazon S3 che contiene il manifest per la partizione.

Il formato DDL per aggiungere le partizioni è il seguente.

```
ALTER TABLE tbl_name
ADD IF NOT EXISTS PARTITION(pcolumn1=pvalue1[,...])
LOCATION
's3://s3-bucket/prefix/_symlink_format_manifest/partition-path'
```

Oppure eseguire la DDL che punta direttamente al file manifest di Delta Lake.

```
ALTER TABLE tbl_name
ADD IF NOT EXISTS PARTITION(pcolumn1=pvalue1[,...])
LOCATION
's3://s3-bucket/prefix/_symlink_format_manifest/partition-path/manifest'
```

Limitazioni e risoluzione dei problemi per le tabelle Delta Lake

Quando si eseguono query sulle tabelle Delta Lake da Redshift Spectrum, tenere presente quanto segue:

- Se un manifest punta a uno snapshot o una partizione che non esiste più, le query hanno esito negativo fino a quando non viene generato un nuovo manifest valido. Ad esempio, questo potrebbe derivare da un'operazione VACUUM sulla tabella sottostante,
- I manifest di Delta Lake forniscono coerenza solo a livello di partizione.

Nella tabella seguente vengono illustrati alcuni potenziali motivi per alcuni errori quando si esegue una query su una tabella Delta Lake.

Messaggio di errore	Motivo possibile
Il manifest di Delta Lake nel bucket s3-bucket-1 non può contenere voci nel bucket s3-bucket-2.	Le voci del manifest puntano a file in un bucket Amazon S3 diverso da quello specificato.

Messaggio di errore	Motivo possibile
I file Delta Lake dovrebbero trovarsi nella stessa cartella.	Le voci del manifest puntano a file che hanno un prefisso Amazon S3 diverso da quello specificato.
Il file nomefile elencato nel manifest Delta Lake percorso manifest non è stato trovato.	Un file elencato nel manifest non è stato trovato in Amazon S3.
Errore durante il recupero del manifesto Delta Lake.	Il manifest non è stato trovato in Amazon S3.
Percorso S3 non valido.	Una voce nel file manifest non è un percorso Amazon S3 valido o il file manifest è stato danneggiato.

Utilizzo delle tabelle Apache Iceberg con Amazon Redshift

È possibile utilizzare Redshift Spectrum o Redshift Serverless per interrogare le tabelle di Apache Iceberg catalogate in AWS Glue Data Catalog. Apache Iceberg è un formato di tabella open source per data lake. Per ulteriori informazioni, consulta Apache Iceberg nella documentazione di Apache Iceberg.

Amazon Redshift fornisce coerenza transazionale per l'interrogazione delle tabelle Apache Iceberg. Puoi manipolare i dati nelle tue tabelle utilizzando servizi conformi ad ACID (atomicità, consistenza, isolamento, durabilità) come Amazon Athena e Amazon EMR mentre esegui query utilizzando Amazon Redshift. Amazon Redshift può utilizzare le statistiche delle tabelle archiviate nei metadati di Apache Iceberg per ottimizzare i piani di query e ridurre le scansioni dei file durante l'elaborazione delle query. Con Amazon Redshift SQL, puoi unire tabelle Redshift con tabelle data lake.

Per iniziare a usare le tabelle Iceberg con Amazon Redshift:

1. Crea una tabella Apache Iceberg su un AWS Glue Data Catalog database utilizzando un servizio compatibile come Amazon Athena o Amazon EMR. Per creare una tabella Iceberg usando Athena, vedi Utilizzo delle tabelle Apache Iceberg nella Guida per l'utente di Amazon Athena.
2. Crea un cluster Amazon Redshift o un gruppo di lavoro Redshift Serverless con un ruolo IAM associato che consenta l'accesso al tuo data lake. Per informazioni su come creare cluster o

gruppi di lavoro, consulta Cluster forniti da Amazon Redshift e Redshift Serverless nella Guida alle operazioni di base di Amazon Redshift.

3. Connettiti al tuo cluster o gruppo di lavoro utilizzando l'editor di query v2 o un client SQL di terze parti. Per informazioni su come connettersi utilizzando l'editor di query v2, consulta Connessione a un data warehouse Amazon Redshift utilizzando gli strumenti client SQL nella Amazon Redshift Management Guide.
4. Crea uno schema esterno nel database Amazon Redshift per uno specifico database del Catalogo dati che include le tabelle Iceberg. Per informazioni sulla creazione di uno schema esterno, consulta Creazione di schemi esterni per Amazon Redshift Spectrum.
5. Esegui query SQL per accedere alle tabelle Iceberg nello schema esterno che hai creato.

Considerazioni sull'utilizzo delle tabelle Apache Iceberg con Amazon Redshift

Quando utilizzi Amazon Redshift con le tabelle Iceberg, considera quanto segue:

- Supporto per la versione Iceberg: Amazon Redshift supporta l'esecuzione di query sulle seguenti versioni delle tabelle Iceberg:
  - Versione 1 che definisce la modalità di gestione delle tabelle analitiche di grandi dimensioni utilizzando file di dati immutabili.
  - Versione 2 che aggiunge la possibilità di supportare l'aggiornamento e l'eliminazione a livello di riga mantenendo invariati i file di dati esistenti e gestendo le modifiche ai dati della tabella utilizzando i file di eliminazione.

  Per la differenza tra le tabelle v1 e v2, consulta Modifiche al tipo di formato nella documentazione di Apache Iceberg.
- Solo query: Amazon Redshift supporta l'accesso in sola lettura alle tabelle Apache Iceberg. Supporta query di selezione coerenti a livello transazionale. Puoi utilizzare un servizio come Amazon Athena per definire e aggiornare lo schema delle tabelle Iceberg in AWS Glue Data Catalog.
- Aggiungere partizioni: non è necessario aggiungere manualmente le partizioni per le tabelle Apache Iceberg. Le nuove partizioni nelle tabelle Apache Iceberg vengono rilevate automaticamente da Amazon Redshift e non è necessaria alcuna operazione manuale per aggiornare le partizioni nella definizione della tabella. Eventuali modifiche alle specifiche della

partizione vengono inoltre applicate automaticamente alle richieste senza alcun intervento da parte dell'utente.

- Inserimento di dati Iceberg in Amazon Redshift: puoi utilizzare i comandi INSERT INTO o CREATE TABLE AS per importare dati dalla tua tabella Iceberg in una tabella Amazon Redshift locale. Al momento non è possibile utilizzare il comando COPY per inserire il contenuto di una tabella Apache Iceberg in una tabella Amazon Redshift locale.
- Viste materializzate: puoi creare le viste materializzate nelle tabelle Apache Iceberg come faresti per qualsiasi altra tabella esterna in Amazon Redshift. Le stesse considerazioni per altri formati di tabelle di data lake si applicano alle tabelle Apache Iceberg. Gli aggiornamenti incrementali, gli aggiornamenti automatici, la riscrittura automatica delle query e gli MV automatici sulle tabelle dei data lake non sono attualmente supportati.
- AWS Lake Formation controllo granulare degli accessi: Amazon Redshift supporta il controllo AWS Lake Formation granulare degli accessi sulle tabelle Apache Iceberg.
- Parametri di gestione dei dati definiti dall'utente: Amazon Redshift supporta parametri di gestione dei dati definiti dall'utente nelle tabelle Apache Iceberg. Si utilizzano parametri di gestione dei dati definiti dall'utente sui file esistenti per personalizzare i dati interrogati in tabelle esterne ed evitare errori di scansione. Questi parametri forniscono funzionalità per gestire le discrepanze tra lo schema della tabella e i dati effettivi sui file. È possibile utilizzare parametri di gestione dei dati definiti dall'utente anche nelle tabelle Apache Iceberg.
- Condivisione dei dati: la condivisione dei dati di Amazon Redshift attualmente non supporta le tabelle data lake, incluse le tabelle Apache Iceberg.
- Domande sui viaggi nel tempo: le query sui viaggi nel tempo non sono attualmente supportate con le tabelle Apache Iceberg.
- Prezzi: quando accedi alle tabelle Iceberg da un cluster, ti vengono addebitati i prezzi di Redshift Spectrum. Quando accedi alle tabelle Iceberg da un gruppo di lavoro, ti vengono addebitati i prezzi di Redshift serverless. Per ulteriori informazioni sui prezzi di Redshift Spectrum e Redshift Serverless, consulta Prezzi di Amazon Redshift.

Argomenti

Tipi di dati supportati con le tabelle Apache Iceberg

Amazon Redshift può eseguire query su tabelle Iceberg che contengono i seguenti tipi di dati:

```
binary
boolean
date
decimal
double
float
int
list
long
map
string
struct
timestamp without time zone
```

Per ulteriori informazioni sui tipi di tabella Iceberg, consulta Schemi per Iceberg nella documentazione di Apache.

La tabella riportata di seguito mostra la relazione tra i tipi di dati Amazon Redshift e i tipi di dati della tabella Iceberg.

Tipo Iceberg	Tipo di Amazon Redshift	Note
`boolean`	`boolean`	
-	`tinyint`	Non supportato per le tabelle Iceberg in Amazon Redshift.
-	`smallint`	Non supportato per le tabelle Iceberg in Amazon Redshift.
`int`	`int`	Nelle istruzioni SQL di Amazon Redshift, questo tipo è `INTEGER`.
`long`	`bigint`	
`double`	`double`	
`float`	`float`	
`decimal(P, S)`	`decimal(P, S)`	`P` è precisione, `S` è scala.

Tipo Iceberg	Tipo di Amazon Redshift	Note
-	`char`	Non supportato per le tabelle Iceberg in Amazon Redshift.
`string`	`string`	Nelle istruzioni SQL di Amazon Redshift, questo tipo è `VARCHAR`.
`binary`	`binary`	
`date`	`date`	
`time`	-	
`timestamp`	`timestamp`	
`timestamptz`	-	Il tipo timestamptz non è attualmente supportato in Redshift Spectrum.
`list<E>`	`array`	
`map<K,V>`	`map`	
`struct<...>`	`struct`	
`fixed(L)`	-	Il tipo `fixed(L)` non è attualmente supportato in Redshift Spectrum.

Per ulteriori informazioni sui tipi di dati di Amazon Redshift, consultare Tipi di dati.

Miglioramento delle prestazioni delle query di Amazon Redshift Spectrum

Esaminare il piano di query per determinare quali fasi sono state inviate al livello di Amazon Redshift Spectrum.

Le fasi seguenti sono correlate alla query di Redshift Spectrum:

- S3 Seq Scan
- S3 HashAggregate
- S3 Query Scan
- Seq Scan PartitionInfo
- Partition Loop

Il seguente esempio mostra un piano di query per una query che unisce in join una tabella esterna a una tabella locale. Prendi nota dei HashAggregate passaggi S3 Seq Scan e S3 eseguiti sui dati su Amazon S3.

```
explain
select top 10 spectrum.sales.eventid, sum(spectrum.sales.pricepaid)
from spectrum.sales, event
where spectrum.sales.eventid = event.eventid
and spectrum.sales.pricepaid > 30
group by spectrum.sales.eventid
order by 2 desc;
```

```
QUERY PLAN


------------------------------------------------------------------------------
XN Limit  (cost=1001055770628.63..1001055770628.65 rows=10 width=31)


  ->  XN Merge  (cost=1001055770628.63..1001055770629.13 rows=200 width=31)


        Merge Key: sum(sales.derived_col2)


        ->  XN Network  (cost=1001055770628.63..1001055770629.13 rows=200 width=31)


              Send to leader
```

```
             ->  XN Sort  (cost=1001055770628.63..1001055770629.13 rows=200 width=31)

                   Sort Key: sum(sales.derived_col2)

                   ->  XN HashAggregate  (cost=1055770620.49..1055770620.99 rows=200
width=31)

                         ->  XN Hash Join DS_BCAST_INNER  (cost=3119.97..1055769620.49
rows=200000 width=31)

                               Hash Cond: ("outer".derived_col1 = "inner".eventid)

                               ->  XN S3 Query Scan sales  (cost=3010.00..5010.50
rows=200000 width=31)

                                     ->  S3 HashAggregate  (cost=3010.00..3010.50
rows=200000 width=16)

                                           ->  S3 Seq Scan spectrum.sales
location:"s3://redshift-downloads/tickit/spectrum/sales" format:TEXT
(cost=0.00..2150.00 rows=172000 width=16)
                                                 Filter: (pricepaid > 30.00)

                               ->  XN Hash  (cost=87.98..87.98 rows=8798 width=4)

                                     ->  XN Seq Scan on event  (cost=0.00..87.98
rows=8798 width=4)
```

Nota i seguenti elementi nel piano di query:

- Il nodo `S3 Seq Scan` mostra che il filtro `pricepaid > 30.00` è stato elaborato nel livello di Redshift Spectrum.

  Un nodo di filtro sotto il nodo `XN S3 Query Scan` indica l'elaborazione di predicati in Amazon Redshift sopra i dati restituiti dal livello di Redshift Spectrum.
- Il nodo `S3 HashAggregate` indica l'aggregazione nel livello Redshift Spectrum per il gruppo mediante la clausola (`group by spectrum.sales.eventid`).

Per migliorare le prestazioni di Redshift Spectrum procedi come segue:

- Utilizza file di dati in formato Apache Parquet. Parquet archivia i dati in un formato a colonne, di conseguenza Redshift Spectrum può eliminare le colonne non necessarie dalla scansione. Se i dati sono in un formato file di testo, Redshift Spectrum deve eseguire la scansione dell'intero file.
- Utilizza più file per ottimizzare l'elaborazione parallela. Fai in modo che le dimensioni dei file siano superiori a 64 MB. Evita l'asimmetria nella dimensione dei dati mantenendo i file all'incirca della stessa dimensione. Per informazioni sui file di Apache Parquet e consigli di configurazione, consulta Formato dei file: configurazioni nella documentazione di Apache Parquet.
- Utilizza il minor numero di colonne possibile nelle query.
- Inserire le tabelle dei fatti di grandi dimensioni in Amazon S3 e le tabelle delle dimensioni più piccole utilizzate di frequente nel database Amazon Redshift locale.
- Aggiorna le statistiche delle tabelle esterne impostando il parametro TABLE PROPERTIES numRows. Utilizzare CREATE EXTERNAL TABLE o ALTER TABLE per impostare il parametro TABLE PROPERTIES numRows allo scopo di riflettere il numero di righe nella tabella. Amazon Redshift non analizza le tabelle esterne per generare le statistiche delle tabelle che l'ottimizzatore di query utilizza per generare un piano di query. Se le statistiche della tabella non sono impostate per una tabella esterna, Amazon Redshift genera un piano di esecuzione della query. Amazon Redshift genera questo piano in base al presupposto che le tabelle esterne sono le tabelle più grandi e che quelle locali sono le più piccole.
- Il pianificatore di query di Amazon Redshift, quando possibile, trasmette i predicati e le aggregazioni al livello di query di Redshift Spectrum. Quando da Amazon S3 sono restituite grandi quantità di dati, l'elaborazione è limitata dalle risorse del cluster. Redshift Spectrum esegue un dimensionamento automatico per elaborare le richieste di grandi dimensioni. Di conseguenza, le prestazioni globali migliorano ogni volta che trasmetti l'elaborazione al livello di Redshift Spectrum.
- Scrivi le query per utilizzare filtri e aggregazioni che possono essere trasmessi al livello di Redshift Spectrum.

  Di seguito sono riportati esempi di alcune operazioni che possono essere trasmesse al livello di Redshift Spectrum:

  - Clausole GROUP BY
  - Condizioni di confronto e di corrispondenza di modelli, come LIKE.
  - Funzioni di aggregazione come COUNT, SUM, AVG, MIN e MAX.
  - Funzioni di stringa.

Le operazioni che non possono essere trasmesse al livello di Redshift Spectrum includono DISTINCT e ORDER BY.

- Utilizza le partizioni per limitare i dati sottoposti a scansione. Partiziona i dati in base ai predicati di query più comuni, quindi riduci le partizioni filtrandone le colonne. Per ulteriori informazioni, consultare Partizionamento delle tabelle esterne di Redshift Spectrum.

  Eseguire una query su SVL_S3PARTITION per visualizzare le partizioni totali e quelle qualificate.
- Utilizza AWS Glue il generatore di statistiche per calcolare le statistiche a livello di colonna per le tabelle. AWS Glue Data Catalog Una volta AWS Glue generate le statistiche per le tabelle nel catalogo dati, Amazon Redshift Spectrum utilizza automaticamente tali statistiche per ottimizzare il piano di query. Per ulteriori informazioni sull'utilizzo delle statistiche a livello di colonna AWS Glue, consulta Working with column statistics nella Developer Guide.AWS Glue

Impostazione delle opzioni di gestione dati

È possibile impostare i parametri della tabella quando si creano tabelle esterne per personalizzare i dati sottoposti a query nelle tabelle esterne. In caso contrario, si possono verificare errori di scansione. Per ulteriori informazioni su queste proprietà, consultare PROPRIETÀ DI TABELLA in CREATE EXTERNAL TABLE. Per alcuni esempi, consulta Esempi di gestione dei dati. Per un elenco dei possibili errori, consultare SVL_SPECTRUM_SCAN_ERROR

È possibile impostare le seguenti PROPRIETÀ DI TABELLA quando si creano tabelle esterne per specificare la gestione dell'input per i dati sottoposti a query in tabelle esterne.

- `column_count_mismatch_handling` consente di determinare se il file contiene più o meno valori per una riga rispetto al numero di colonne specificato nella definizione della tabella esterna.
- `invalid_char_handling` per specificare la gestione dell'input per i caratteri non validi nelle colonne contenenti dati VARCHAR, CHAR e stringa. Quando si specifica REPLACE per `invalid_char_handling`, è possibile specificare il carattere sostitutivo da utilizzare.
- `numeric_overflow_handling` per specificare la gestione dell'eccedenza cast in colonne contenenti dati interi e decimali.
- `surplus_bytes_handling` per specificare la gestione degli input per i byte in eccesso nelle colonne contenenti dati VARBYTE.
- `surplus_char_handling` per specificare la gestione dell'input per i caratteri eccedenti nelle colonne contenenti dati VARCHAR, CHAR e stringa.

È possibile impostare un'opzione di configurazione per annullare le query che superano il numero massimo di errori. Per ulteriori informazioni, consulta spectrum_query_maxerror.

Esempio: esecuzione di sottoquery correlate in Redshift Spectrum

È possibile eseguire sottoquery correlate in Redshift Spectrum. La pseudocolonna `$spectrum_oid` consente di eseguire query correlate con Redshift Spectrum. Per eseguire una sottoquery correlata, la pseudocolonna `$spectrum_oid` deve essere abilitata ma non viene visualizzata nell'istruzione SQL. Per ulteriori informazioni, consulta Pseudocolonne.

Per creare lo schema esterno e le tabelle esterne per questo esempio, consulta Nozioni di base su Amazon Redshift Spectrum.

Di seguito è riportato un esempio di sottoquery correlata in Redshift Spectrum.

```
select *
from myspectrum_schema.sales s
where exists
( select *
from myspectrum_schema.listing l
where l.listid = s.listid )
order by salesid
limit 5;
```

```
salesid  listid   sellerid   buyerid   eventid   dateid  qtysold  pricepaid
 commission   saletime
1        1        36861      21191     7872      1875    4        728         109.2
    2008-02-18 02:36:48
2        4        8117       11498     4337      1983    2        76          11.4
    2008-06-06 05:00:16
3        5        1616       17433     8647      1983    2        350         52.5
    2008-06-06 08:26:17
4        5        1616       19715     8647      1986    1        175         26.25
    2008-06-09 08:38:52
5        6        47402      14115     8240      2069    2        154         23.1
    2008-08-31 09:17:02
```

Monitoraggio dei parametri in Amazon Redshift Spectrum

È possibile monitorare le query di Amazon Redshift Spectrum utilizzando le seguenti viste di sistema:

- SVL_S3QUERY

  Utilizza la vista SVL_S3QUERY per ottenere dettagli riguardanti le query di Redshift Spectrum (query S3) a livello di segmento e di sezione di nodo.
- SVL_S3QUERY_SUMMARY

  Utilizzare la vista SVL_S3QUERY_SUMMARY per ottenere un riepilogo di tutte le query Amazon Redshift Spectrum (query S3) che sono state eseguite nel sistema.

Di seguito sono elencati alcuni elementi da cercare in SVL_S3QUERY_SUMMARY:

- Il numero di file elaborati dalla query di Redshift Spectrum.
- Il numero di byte sottoposti a scansione da Amazon S3. Il costo di una query di Redshift Spectrum viene riflesso nella quantità di dati sottoposti a scansione da Amazon S3.
- Il numero di byte restituiti dal livello Redshift Spectrum al cluster. Se viene restituita una grande quantità di dati, è possibile che le prestazioni del sistema peggiorino.
- La durata massima e media delle richieste Redshift Spectrum. Le richieste di lunga durata potrebbero indicare un collo di bottiglia.

Risoluzione dei problemi relativi alle query in Amazon Redshift Spectrum

Di seguito, è possibile trovare un riferimento rapido che identifica e risolve alcuni problemi comuni che si possono incontrare con le query di Amazon Redshift Spectrum. Per visualizzare gli errori generati dalle query di Redshift Spectrum, eseguire una query sulla tabella di sistema SVL_S3LOG.

Argomenti

- Superamento del numero di nuovi tentativi
- Accesso limitato
- Superamento del limite di risorse
- Nessuna riga restituita per una tabella partizionata

- Errore non autorizzato
- Formati di dati non compatibili
- Errore di sintassi durante l'utilizzo della DDL Hive in Amazon Redshift
- Autorizzazione per creare tabelle temporanee
- Intervallo non valido
- Numero versione di Parquet non valido

Superamento del numero di nuovi tentativi

In caso di timeout di una richiesta di Amazon Redshift Spectrum, la richiesta viene annullata e inoltrata di nuovo. Dopo cinque tentativi non riusciti, la query non riesce con il seguente errore.

```
error:  Spectrum Scan Error: Retries exceeded
```

Le cause possibili sono:

- File di grandi dimensioni (superiori a 1 GB). Verificare la dimensione dei file in Amazon S3 e cercare file di dimensioni voluminose e disuguali. Dividi i file di grandi dimensioni in file più piccoli, di dimensione tra 100 MB e 1 GB. Fai in modo che i file siano della stessa dimensione.
- Throughput di rete lento. Riesegui la query in seguito.

Accesso limitato

Amazon Redshift Spectrum è soggetto alle quote di servizio di AWS altri servizi. In caso di utilizzo elevato, le richieste Redshift Spectrum potrebbero rallentare, causando il seguente errore.

```
error:  Spectrum Scan Error: Access throttled
```

Possono verificarsi due tipi di limitazione:

- Accesso limitato da Amazon S3
- Accesso limitato da. AWS KMS

Il contesto di errore fornisce ulteriori dettagli sul tipo di limitazione. Di seguito, puoi trovare le cause e le possibili risoluzioni per questa limitazione.

Accesso limitato da Amazon S3

Amazon S3 potrebbe limitare una richiesta Redshift Spectrum se la frequenza di richieste di lettura su un prefisso è troppo alta. Per informazioni sulla frequenza di richieste GET/HEAD che si può ottenere in Amazon S3, consultare Ottimizzazione delle prestazioni di Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service. La frequenza di richieste GET/HEAD di Amazon S3 tiene conto di tutte le richieste GET/HEAD su un prefisso, quindi diverse applicazioni che accedono allo stesso prefisso condividono la frequenza totale delle richieste.

Se le richieste di Redshift Spectrum vengono spesso limitate da Amazon S3, che riduce il numero di richieste GET/HEAD di Amazon S3 che Redshift Spectrum effettua ad Amazon S3. A tale scopo, prova a unire file di piccole dimensioni in file più grandi. Ti consigliamo di utilizzare dimensioni pari a 64 MB o superiori.

Considerare inoltre la possibilità di partizionare le tabelle Redshift Spectrum per beneficiare del filtraggio anticipato e ridurre il numero di file a cui si accede in Amazon S3. Per ulteriori informazioni, consultare Partizionamento delle tabelle esterne di Redshift Spectrum.

Accesso limitato da AWS KMS

Se memorizzi i dati in Amazon S3 utilizzando la crittografia lato server (SSE-S3 o SSE-KMS), Amazon S3 richiama un'operazione API per ogni file a cui accede Redshift Spectrum. AWS KMS Queste richieste vengono conteggiate per la quota delle operazioni di crittografia; per ulteriori informazioni, consultare Quote di richiesta AWS KMS. Per ulteriori informazioni su SSE-S3 e SSE-KMS, consultare Protezione dei dati tramite crittografia lato server e Protezione dei dati tramite crittografia lato server con Chiavi KMS archiviate in AWS KMS nella Guida per l'utente di Amazon Simple Storage Service.

Un primo passo per ridurre il numero di richieste inviate da Redshift Spectrum AWS KMS consiste nel ridurre il numero di file a cui si accede. A tale scopo, prova a unire file di piccole dimensioni in file più grandi. Ti consigliamo di utilizzare dimensioni pari a 64 MB o superiori.

Se le tue richieste Redshift Spectrum vengono spesso limitate AWS KMS, prendi in considerazione la possibilità di richiedere un aumento della quota per la frequenza di AWS KMS richieste per le operazioni crittografiche. Per richiedere un aumento della quota, consulta Limiti del servizio AWS in Riferimenti generali di Amazon Web Services.

Superamento del limite di risorse

Redshift Spectrum applica un limite superiore sulla quantità di memoria che una richiesta può utilizzare. Una richiesta Redshift Spectrum che richiede più memoria non riesce, causando il seguente errore.

```
error:  Spectrum Scan Error: Resource limit exceeded
```

Esistono due motivi comuni che possono causare il superamento dei limiti di memoria di una richiesta Redshift Spectrum:

- Redshift Spectrum elabora una grande porzione di dati che non possono essere suddivisi in blocchi più piccoli.
- Un passaggio di aggregazione di grandi dimensioni viene elaborato da Redshift Spectrum.

Ti consigliamo di utilizzare un formato di file che supporti le letture parallele con dimensioni di divisione pari o inferiori a 128 MB. consultare Creazione di file di dati per le query in Amazon Redshift Spectrum per i formati di file supportati e le linee guida generiche per la creazione di file di dati. Quando utilizzi formati di file o algoritmi di compressione che non supportano le letture parallele, ti consigliamo di mantenere le dimensioni dei file tra 64 MB e 128 MB.

Nessuna riga restituita per una tabella partizionata

Se la query restituisce zero righe da una tabella esterna partizionata, verifica se una partizione è stata aggiunta per questa tabella esterna. Redshift Spectrum esegue la scansione dei file in una posizione Amazon S3 che è stata esplicitamente aggiunta utilizzando `ALTER TABLE … ADD PARTITION`. Esegui la query sulla vista SVV_EXTERNAL_PARTITIONS per trovare le partizioni esistenti. Esegui `ALTER TABLE … ADD PARTITION` per ogni partizione mancante.

Errore non autorizzato

Verificare che il ruolo IAM per il cluster consenta l'accesso agli oggetti dei file Amazon S3. Se il database esterno si trova in Amazon Athena, verificare che il ruolo IAM consenta l'accesso alle risorse Athena. Per ulteriori informazioni, consultare Policy IAM per Amazon Redshift Spectrum.

Formati di dati non compatibili

Per un formato di file a colonne, come Apache Parquet, il tipo di colonna è incorporato ai dati. Il tipo di colonna nella definizione CREATE EXTERNAL TABLE deve corrispondere al tipo di colonna del file di dati. In caso di mancata corrispondenza, viene visualizzato un messaggio di errore simile al seguente:

```
File 'https://s3bucket/location/file has an incompatible Parquet schema
for column ‘s3://s3bucket/location.col1'. Column type: VARCHAR, Par
```

Il messaggio potrebbe essere troncato a causa del limite della lunghezza del messaggio. Per recuperare il messaggio completo, incluso il nome e il tipo di colonna, eseguire una query sulla vista di sistema SVL_S3LOG.

Le seguenti query di esempio SVL_S3LOG per l'ultima query completata.

```
select message
from svl_s3log
where query = pg_last_query_id()
order by query,segment,slice;
```

Quanto segue è un esempio di risultato che mostra un messaggio di errore completo.

```
                            message
-----------------------------------------------------------------------------------------
Spectrum Scan Error. File 'https://s3bucket/location/file has an incompatible
Parquet schema for column ' s3bucket/location.col1'.
Column type: VARCHAR, Parquet schema:\noptional int64 l_orderkey [i:0 d:1 r:0]\n
```

Per correggere l'errore, modifica la tabella esterna affinché corrisponda al tipo di colonna del file Parquet.

Errore di sintassi durante l'utilizzo della DDL Hive in Amazon Redshift

Per CREATE EXTERNAL TABLE, Amazon Redshift supporta un linguaggio DDL che è simile al linguaggio DDL Hive. Tuttavia, i due tipi di DDL non sono sempre esattamente gli stessi. Se si copia il DDL Hive per creare o modificare le tabelle esterne Amazon Redshift, è possibile che si verifichino errori di sintassi. Di seguito sono riportati alcuni esempi di differenze tra Amazon Redshift e la DDL Hive:

- Amazon Redshift richiede l'utilizzo di virgolette singole (') mentre la DDL Hive supporta le virgolette doppie (").
- Amazon Redshift non supporta il tipo di dati STRING. Utilizza invece VARCHAR.

Autorizzazione per creare tabelle temporanee

Per eseguire le query di Redshift Spectrum, l'utente del database deve disporre dell'autorizzazione per creare tabelle temporanee nel database. L'esempio seguente concede l'autorizzazione temporanea per il database `spectrumdb` al gruppo di utenti `spectrumusers`.

```
grant temp on database spectrumdb  to group spectrumusers;
```

Per ulteriori informazioni, consultare GRANT.

Intervallo non valido

Redshift Spectrum prevede che i file in Amazon S3 appartenenti a una tabella esterna non vengano sovrascritti durante una query. In tal caso, può restituire il seguente errore.

```
Error: HTTP response error code: 416 Message: InvalidRange The requested range is not
 satisfiable
```

Per evitare l'errore, assicurati che i file Amazon S3 non vengano sovrascritti mentre sono soggetti a query con Redshift Spectrum.

Numero versione di Parquet non valido

Redshift Spectrum controlla i metadati di ogni file Apache Parquet a cui accede. Se il controllo ha esito negativo, può essere visualizzato un errore simile al seguente:

```
File 'https://s3.region.amazonaws.com/s3bucket/location/file has an invalid version
 number
```

L'esito negativo del controllo può essere causato da due motivi comuni:

- Il file Parquet è stato sovrascritto durante la query (consulta Intervallo non valido).
- Il file Parquet è danneggiato.

Tutorial: Esecuzione di query su dati nidificati con Amazon Redshift Spectrum

Panoramica

Amazon Redshift Spectrum supporta l'esecuzione di query di dati nidificati nei formati di file Parquet, ORC, JSON e Ion. Redshift Spectrum accede ai dati mediante tabelle esterne. Puoi creare tabelle esterne che utilizzano i tipi di dati complessi `struct`, `array` e `map`.

Ad esempio, si supponga che il file di dati contenga i seguenti dati in Amazon S3 all'interno di una cartella denominata `customers`. Anche se non è presente un singolo elemento root, ciascun oggetto JSON in questi dati di esempio rappresenta una riga in una tabella.

```
{"id": 1,
 "name": {"given": "John", "family": "Smith"},
 "phones": ["123-457789"],
 "orders": [{"shipdate": "2018-03-01T11:59:59.000Z", "price": 100.50},
            {"shipdate": "2018-03-01T09:10:00.000Z", "price": 99.12}]
}
{"id": 2,
 "name": {"given": "Jenny", "family": "Doe"},
 "phones": ["858-8675309", "415-9876543"],
 "orders": []
}
{"id": 3,
 "name": {"given": "Andy", "family": "Jones"},
 "phones": [],
 "orders": [{"shipdate": "2018-03-02T08:02:15.000Z", "price": 13.50}]
}
```

Puoi utilizzare Amazon Redshift Spectrum per eseguire query sui dati nidificati nei file. Il seguente tutorial illustra la procedura da seguire con i dati Apache Parquet.

Per i prerequisiti, le fasi e i casi d'uso relativi ai dati nidificati del tutorial, consultare i seguenti argomenti:

- Prerequisiti
- Fase 1: creazione di una tabella esterna contenente dati nidificati
- Fase 2: Esecuzione di query sui dati nidificati in Amazon S3 con estensioni SQL

- Casi d'uso dei dati nidificati
- Limitazioni relative ai dati annidati (anteprima)
- Serializzazione di JSON nidificato complesso

Prerequisiti

Se ancora non utilizzi Redshift Spectrum, segui la procedura in Nozioni di base su Amazon Redshift Spectrum prima di continuare.

Per creare uno schema esterno, sostituire l'ARN del ruolo IAM nel comando seguente con l'ARN del ruolo creato in Creazione di un ruolo IAM. Quindi eseguire il comando nel proprio client SQL.

```
create external schema spectrum
from data catalog
database 'myspectrum_db'
iam_role 'arn:aws:iam::123456789012:role/myspectrum_role'
create external database if not exists;
```

Fase 1: creazione di una tabella esterna contenente dati nidificati

È possibile visualizzare i dati di origine scaricandoli da Amazon S3.

Per creare la tabella esterna per questo tutorial, utilizza il comando seguente.

```
CREATE EXTERNAL TABLE spectrum.customers (
  id     int,
  name   struct<given:varchar(20), family:varchar(20)>,
  phones array<varchar(20)>,
  orders array<struct<shipdate:timestamp, price:double precision>>
)
STORED AS PARQUET
LOCATION 's3://redshift-downloads/tickit/spectrum/customers/';
```

Nell'esempio precedente, la tabella esterna `spectrum.customers` utilizza i tipi di dati `struct` e `array` per definire colonne con dati nidificati. Amazon Redshift Spectrum supporta l'esecuzione di query di dati nidificati nei formati di file Parquet, ORC, JSON e Ion. Il parametro `STORED AS` è `PARQUET` per i file Apache Parquet. Il parametro `LOCATION` deve fare riferimento alla cartella Amazon S3 che contiene i file o i dati nidificati. Per ulteriori informazioni, consulta CREATE EXTERNAL TABLE.

Puoi nidificare tipi `array` e `struct` a qualsiasi livello. Ad esempio, puoi definire una colonna denominata `toparray` come mostrato nell'esempio seguente.

```
toparray array<struct<nestedarray:
         array<struct<morenestedarray:
         array<string>>>>>
```

Puoi inoltre nidificare tipi `struct` come mostrato per la colonna `x` nell'esempio seguente.

```
x struct<a: string,
         b: struct<c: integer,
                   d: struct<e: string>
                  >
        >
```

Fase 2: Esecuzione di query sui dati nidificati in Amazon S3 con estensioni SQL

Redshift Spectrum supporta l'esecuzione di query su tipi complessi `array`, `map` e `struct` mediante estensioni alla sintassi SQL di Amazon Redshift.

Estensione 1: accesso a colonne di struct

Puoi estrarre dati da colonne `struct` utilizzando una notazione punto che concatena i nomi dei campi in percorsi. Ad esempio, la query seguente restituisce il nome e il cognome dei clienti. L'accesso al nome viene eseguito dal percorso lungo `c.name.given`. L'accesso al cognome viene eseguito dal percorso lungo `c.name.family`.

```
SELECT c.id, c.name.given, c.name.family
FROM   spectrum.customers c;
```

La query precedente restituisce i seguenti dati.

```
id | given | family
---|-------|-------
1  | John  | Smith
2  | Jenny | Doe
3  | Andy  | Jones
(3 rows)
```

Uno `struct` può essere una colonna di un altro `struct`, che a sua volta può essere una colonna di un altro `struct`, a qualsiasi livello. I percorsi di accesso alle colonne in `struct` nidificati in modo così profondo possono essere lunghi. Ad esempio, consultare la definizione relativa alla colonna `x` nell'esempio seguente.

```
x struct<a: string,
         b: struct<c: integer,
                   d: struct<e: string>
                  >
        >
```

Puoi accedere ai dati in `e` come `x.b.d.e`.

Estensione 2: matrici estese a una clausola FROM

Puoi estrarre dati da colonne `array` (e, per estensione, da colonne `map`) specificando le colonne `array` in una clausola `FROM` al posto dei nomi delle tabelle. L'estensione si applica alla clausola `FROM` della query principale e, inoltre, alle clausole `FROM` delle query secondarie.

Puoi fare riferimento a elementi `array` in base alla posizione, ad esempio `c.orders[0]` (anteprima).

Combinando `arrays` estese e join, puoi annullare annidamenti in vari modi, come illustrato nei seguenti casi d'uso.

Annullamento di annidamenti mediante inner join

La seguente query seleziona gli ID dei clienti e le date di spedizione degli ordini per i clienti associati a degli ordini. L'estensione SQL nella clausola FROM `c.orders o` dipende dall'alias `c`.

```
SELECT c.id, o.shipdate
FROM   spectrum.customers c, c.orders o
```

Per ciascun cliente `c` associato a degli ordini, la clausola `FROM` restituisce una riga per ogni ordine `o` del cliente `c`. La riga in questione combina la riga del cliente `c` e quella dell'ordine `o`. In seguito, la clausola `SELECT` mantiene solo `c.id` e `o.shipdate`. Il risultato è il seguente.

```
id|      shipdate
--|----------------------
1 |2018-03-01  11:59:59
```

```
1 |2018-03-01  09:10:00
3 |2018-03-02  08:02:15
(3 rows)
```

L'alias `c` fornisce l'accesso ai campi dei clienti, mentre l'alias `o` consente l'accesso ai campi degli ordini.

La semantica è simile a quella di SQL standard. È possibile pensare a un processo in cui la clausola `FROM` esegue il loop nidificato mostrato sotto, mentre `SELECT` sceglie i campi da restituire.

```
for each customer c in spectrum.customers
  for each order o in c.orders
     output c.id and o.shipdate
```

Pertanto, se un cliente non è associato ad alcun ordine, non verrà mostrato tra i risultati.

Puoi pensare anche a un processo in cui la clausola `FROM` esegue un `JOIN` con la tabella `customers` e la matrice `orders`. In effetti, puoi anche scrivere la query, come mostrato nell'esempio seguente.

```
SELECT c.id, o.shipdate
FROM   spectrum.customers c INNER JOIN c.orders o ON true
```

> **Note**
>
> Se è presente uno schema denominato `c` con una tabella chiamata `orders`, allora `c.orders` fa riferimento alla tabella `orders` e non alla colonna della matrice di `customers`.

Annullamento di annidamenti mediante left join

La query seguente restituisce tutti i nomi dei clienti e i loro ordini. Se un cliente non ha effettuato alcun ordine, il suo nome viene comunque restituito. Tuttavia, in questo caso, l'ordine presenta colonne NULL, come mostrato nell'esempio seguente per Jenny Doe.

```
SELECT c.id, c.name.given, c.name.family, o.shipdate, o.price
FROM   spectrum.customers c LEFT JOIN c.orders o ON true
```

La query precedente restituisce i seguenti dati.

```
id  |  given  | family  |    shipdate          | price
----|---------|---------|----------------------|--------
 1  |  John   | Smith   | 2018-03-01  11:59:59 | 100.5
 1  |  John   | Smith   | 2018-03-01  09:10:00 |  99.12
 2  |  Jenny  | Doe     |                      |
 3  |  Andy   | Jones   | 2018-03-02  08:02:15 |  13.5
 (4 rows)
```

Estensione 3: accesso diretto a una matrice di scalari tramite un alias

Quando un alias p in una clausola FROM si estende a una matrice di scalari, la query fa riferimento ai valori di p come p. Ad esempio, la query seguente genera coppie di nomi e numeri di telefono dei clienti.

```
SELECT c.name.given, c.name.family, p AS phone
FROM   spectrum.customers c LEFT JOIN c.phones p ON true
```

La query precedente restituisce i seguenti dati.

```
given  |  family  |  phone
-------|----------|-----------
John   | Smith    | 123-4577891
Jenny  | Doe      | 858-8675309
Jenny  | Doe      | 415-9876543
Andy   | Jones    |
(4 rows)
```

Estensione 4: accesso agli elementi del tipo di dati map

Redshift Spectrum tratta `map` come un tipo di dati `array` che contiene tipi `struct` con una colonna `key` e una colonna `value`. `key` deve essere uno `scalar`; il valore può essere qualunque tipo di dati.

Ad esempio, il codice seguente crea una tabella esterna con un tipo `map` per l'archiviazione dei numeri di telefono.

```
CREATE EXTERNAL TABLE spectrum.customers2 (
  id     int,
  name   struct<given:varchar(20), family:varchar(20)>,
  phones map<varchar(20), varchar(20)>,
  orders array<struct<shipdate:timestamp, price:double precision>>
```

```
)
STORED AS PARQUET
LOCATION 's3://redshift-downloads/tickit/spectrum/customers/';
```

Dal momento che un tipo `map` si comporta come un tipo `array` con colonne `key` e `value`, puoi pensare agli schemi precedenti come a quelli riportati di seguito.

```
CREATE EXTERNAL TABLE spectrum.customers3 (
  id     int,
  name   struct<given:varchar(20), family:varchar(20)>,
  phones array<struct<key:varchar(20), value:varchar(20)>>,
  orders array<struct<shipdate:timestamp, price:double precision>>
)
STORED AS PARQUET
LOCATION 's3://redshift-downloads/tickit/spectrum/customers/';
```

La query seguente restituisce i nomi dei clienti con un numero di cellulare e il numero per ciascun nome. La query sul tipo map viene considerata equivalente a una query su un `array` nidificato di tipi `struct`. La query seguente restituisce dati solo se hai creato la tabella esterna come descritto in precedenza.

```
SELECT c.name.given, c.name.family, p.value
FROM   spectrum.customers c, c.phones p
WHERE  p.key = 'mobile';
```

> **Note**
>
> La `key` per un tipo `map` è una `string` per i tipi di file Ion e JSON.

Casi d'uso dei dati nidificati

Puoi combinare le estensioni descritte in precedenza con le consuete caratteristiche di SQL. I casi d'uso seguenti presentano alcune combinazioni comuni. Questi esempi offrono una dimostrazione dell'utilizzo dei dati nidificati. Non fanno parte del tutorial.

Argomenti

- Inserimento di dati nidificati
- Aggregazione di dati nidificati con query secondarie

- Unione dei dati di Amazon Redshift e dei dati nidificati

Inserimento di dati nidificati

Puoi utilizzare l'istruzione `CREATE TABLE AS` per inserire dati da una tabella esterna contenente tipi di dati complessi. La query seguente estrae tutti i clienti e i relativi numeri di telefono dalla tabella esterna mediante `LEFT JOIN` e li archivia nella tabella `CustomerPhones` di Amazon Redshift.

```
CREATE TABLE CustomerPhones AS
SELECT  c.name.given, c.name.family, p AS phone
FROM    spectrum.customers c LEFT JOIN c.phones p ON true;
```

Aggregazione di dati nidificati con query secondarie

Puoi utilizzare una query secondaria per aggregare dati nidificati. Questo approccio viene descritto nell'esempio seguente.

```
SELECT c.name.given, c.name.family, (SELECT COUNT(*) FROM c.orders o) AS ordercount
FROM   spectrum.customers c;
```

Vengono restituiti i seguenti dati.

```
given   |  family   |  ordercount
--------|-----------|--------------
 Jenny  |  Doe      |       0
 John   |  Smith    |       2
 Andy   |  Jones    |       1
 (3 rows)
```

> **Note**
>
> Quando aggreghi dati nidificati raggruppandoli in base alla riga padre, il modo più efficiente di procedere è quello illustrato nell'esempio precedente. Nell'esempio in questione, le righe nidificate di `c.orders` sono raggruppate in base alla riga padre `c`. In alternativa, se sai che l'`id` è univoco per ciascun `customer` e `o.shipdate` non è mai null, puoi eseguire l'aggregazione come illustrato nell'esempio seguente. Tuttavia, in genere questo approccio è meno efficiente rispetto a quello dell'esempio precedente.

```
SELECT    c.name.given, c.name.family, COUNT(o.shipdate) AS ordercount
FROM      spectrum.customers c LEFT JOIN c.orders o ON true
GROUP BY  c.id, c.name.given, c.name.family;
```

Puoi inoltre scrivere la query utilizzando una query secondaria nella clausola FROM che fa riferimento a un alias (c) della query con predecessore e che estrae i dati della matrice. L'esempio seguente mostra questo approccio.

```
SELECT c.name.given, c.name.family, s.count AS ordercount
FROM   spectrum.customers c, (SELECT count(*) AS count FROM c.orders o) s;
```

Unione dei dati di Amazon Redshift e dei dati nidificati

È possibile unire i dati di Amazon Redshift e i dati nidificati in una tabella esterna. Ad esempio, si supponga di avere i seguenti dati nidificati in Amazon S3.

```
CREATE EXTERNAL TABLE spectrum.customers2 (
  id      int,
  name    struct<given:varchar(20), family:varchar(20)>,
  phones  array<varchar(20)>,
  orders  array<struct<shipdate:timestamp, item:int>>
);
```

Si supponga inoltre di avere la seguente tabella in Amazon Redshift.

```
CREATE TABLE prices (
  id int,
  price double precision
);
```

La query riportata di seguito restituisce il numero totale e l'importo degli acquisti di ciascun cliente sulla base di quanto indicato in precedenza. L'esempio seguente viene utilizzato soltanto a scopo illustrativo. Restituisce dati solo se hai creato le tabelle descritte in precedenza.

```
SELECT   c.name.given, c.name.family, COUNT(o.date) AS ordercount, SUM(p.price) AS
 ordersum
FROM     spectrum.customers2 c, c.orders o, prices p ON o.item = p.id
GROUP BY c.id, c.name.given, c.name.family;
```

Limitazioni relative ai dati annidati (anteprima)

> **Note**
>
> Le limitazioni contrassegnate con "(anteprima)" nell'elenco seguente si applicano solo ai cluster di anteprima e ai gruppi di lavoro di anteprima creati nelle seguenti regioni.
>
> - Stati Uniti orientali (Ohio) (us-east-2)
> - Stati Uniti orientali (Virginia settentrionale) (us-east-1)
> - Stati Uniti occidentali (California settentrionale) (us-west-1)
> - Asia Pacifico (Tokyo) (ap-northeast-1)
> - Europa (Irlanda) (eu-west-1)
> - Europa (Stoccolma) (eu-north-1)
>
> Per informazioni sulla configurazione di cluster di anteprima, consulta Creazione di un cluster di anteprima nella Guida alla gestione di Amazon Redshift. Per informazioni sulla configurazione dei gruppi di lavoro di anteprima, consulta Creazione di un gruppo di lavoro di anteprima nella Guida alla gestione di Amazon Redshift.

Le seguenti limitazioni si applicano ai dati nidificati:

- Un tipo `array` o `map` può contenere altri tipi `array` o `map`, purché le query su `arrays` o `maps` annidati non restituiscano valori `scalar` (anteprima).
- Amazon Redshift Spectrum supporta i tipi di dati complessi solo come tabelle esterne.
- Le colonne dei risultati delle query secondarie devono essere di primo livello (anteprima).
- Se un'espressione `OUTER JOIN` si riferisce a una tabella nidificata, può fare riferimento solo a quella tabella e alle relative matrici (e mappe) nidificate. Se un'espressione `OUTER JOIN` non si riferisce a una tabella nidificata, può fare riferimento a qualsiasi numero di tabelle non nidificate.
- Se una clausola `FROM` in una query secondaria si riferisce a una tabella nidificata, non può fare riferimento a nessun'altra tabella.
- Se una query secondaria dipende da una tabella nidificata che si riferisce a una tabella padre, la query secondaria può utilizzare la tabella padre solo nella clausola `FROM`. Non puoi utilizzare l'elemento padre in un'altra clausola, ad esempio una di tipo `SELECT` o `WHERE`. Ad esempio, la

seguente query non viene eseguita perché la clausola SELECT della query secondaria si riferisce alla tabella padre c.

```
SELECT c.name.given
FROM   spectrum.customers c
WHERE (SELECT COUNT(c.id) FROM c.phones p WHERE p LIKE '858%') > 1;
```

La query riportata sotto viene eseguita perché l'elemento c è utilizzato solo nella clausola FROM della query secondaria.

```
SELECT c.name.given
FROM   spectrum.customers c
WHERE (SELECT COUNT(*) FROM c.phones p WHERE p LIKE '858%') > 1;
```

- Una query secondaria che accede ai dati nidificati non dalla clausola FROM deve restituire un unico valore. Fanno eccezione solo gli operatori (NOT) EXISTS in una clausola WHERE.
- (NOT) IN non è supportato.
- La profondità di nidificazione massima per tutti i tipi nidificati è 100. Questa restrizione si applica a tutti i formati di file (Parquet, ORC, Ion e JSON).
- Le query secondarie di aggregazione che accedono a dati nidificati possono fare riferimento solo ad arrays e maps nella loro clausola FROM, non a una tabella esterna.
- L'esecuzione di query sulle pseudocolonne di dati nidificati in una tabella di Redshift Spectrum non è supportata. Per ulteriori informazioni, consulta Pseudocolonne.
- Quando si estraggono i dati da colonne di matrici o mappe specificate in una clausola FROM, è possibile selezionare i valori dalle colonne solo se i valori sono scalar. Ad esempio, le seguenti query tentano entrambe di eseguire SELECT per gli elementi dall'interno di una matrice. La query che seleziona arr.a riesce perché arr.a è un valore scalar. La seconda query non riesce perché array è una matrice estratta da s3.nested table nella clausola FROM (anteprima).

```
SELECT array_column FROM s3.nested_table;

array_column
-----------------
[{"a":1},{"b":2}]

SELECT arr.a FROM s3.nested_table t, t.array_column arr;

arr.a
```

```
-----
1

--This query fails to run.
SELECT array FROM s3.nested_table tab, tab.array_column array;
```

Non è possibile utilizzare una matrice o una mappa nella clausola `FROM` che a sua volta proviene da un'altra matrice o mappa. Per selezionare matrici o altre strutture complesse annidate all'interno di altre matrici, prendi in considerazione di usare gli indici nell'istruzione `SELECT`.

Serializzazione di JSON nidificato complesso

Un'alternativa ai metodi illustrati in questo tutorial consiste nell'eseguire una query sulle colonne di raccolta nidificate di primo livello come JSON serializzato. È possibile utilizzare la serializzazione per ispezionare, convertire e importare dati nidificati come JSON con Redshift Spectrum. Questo metodo è supportato per i formati ORC, JSON, Ion e Parquet. Utilizzare il parametro di configurazione della sessione `json_serialization_enable` per configurare il comportamento di serializzazione. Se impostato, i tipi di dati JSON complessi sono serializzati in VARCHAR(65535). È possibile accedere al JSON nidificato con Funzioni JSON. Per ulteriori informazioni, consultare json_serialization_enable.

Ad esempio, senza impostare `json_serialization_enable`, le seguenti query che accedono direttamente alle colonne nidificate avranno esito negativo.

```
SELECT * FROM spectrum.customers LIMIT 1;

=> ERROR:  Nested tables do not support '*' in the SELECT clause.

SELECT name FROM spectrum.customers LIMIT 1;

=> ERROR:  column "name" does not exist in customers
```

L'impostazione di `json_serialization_enable` consente di eseguire le query direttamente sulle raccolte di primo livello.

```
SET json_serialization_enable TO true;

SELECT * FROM spectrum.customers order by id LIMIT 1;

id | name                              | phones         | orders
```

```
---+-------------------------------------+---------------
+-----------------------------------------------------------------------------------------------
1  | {"given": "John", "family": "Smith"} | ["123-457789"] | [{"shipdate":
 "2018-03-01T11:59:59.000Z", "price": 100.50}, {"shipdate": "2018-03-01T09:10:00.000Z",
 "price": 99.12}]

SELECT name FROM spectrum.customers order by id LIMIT 1;

name
---------
{"given": "John", "family": "Smith"}
```

Considerare quanto segue durante la serializzazione del JSON nidificato.

- Quando le colonne di raccolta sono serializzate come VARCHAR(65535), non è possibile accedere direttamente ai relativi sottocampi nidificati come parte della sintassi della query (ad esempio, nella clausola filter). Tuttavia, le funzioni JSON possono essere utilizzate per accedere al JSON nidificato.
- Le seguenti rappresentazioni specializzate non sono supportate:
  - Unioni ORC
  - Mappe ORC con chiavi di tipo complesso
  - Datagrammi Ion
  - SEXP Ion
- I timestamp vengono restituiti come stringhe serializzate ISO.
- Le chiavi delle mappe primitive sono promosse a stringa (ad esempio, da 1 a "1").
- I valori nulli di primo livello vengono serializzati come NULL.
- Se la serializzazione eccede la dimensione massima VARCHAR di 65535, la cella viene impostata su NULL.

Serializzazione di tipi complessi contenenti stringhe JSON

Per impostazione predefinita, i valori stringa contenuti nelle raccolte nidificate vengono serializzati come stringhe JSON con escape. L'escape potrebbe essere indesiderato se le stringhe sono JSON valide. È invece possibile voler scrivere sottoelementi o campi nidificati che sono VARCHAR direttamente come JSON. Abilitare questo comportamento con la configurazione a livello di sessione `json_serialization_parse_nested_strings`. Quando sono impostati sia `json_serialization_enable` che `json_serialization_parse_nested_strings`,

i valori JSON validi sono serializzati in linea senza caratteri escape. Se il valore non è JSON valido, viene eseguito l'escape come se il valore di configurazione `json_serialization_parse_nested_strings` non fosse impostato. Per ulteriori informazioni, consultare json_serialization_parse_nested_strings.

Ad esempio, si supponga che i dati dell'esempio precedente contengano JSON come tipo complesso `structs`nel campo `name` VARCHAR(20):

```
name
---------
{"given": "{\"first\":\"John\",\"middle\":\"James\"}", "family": "Smith"}
```

Quando è impostato `json_serialization_parse_nested_strings`, la colonna `name` viene serializzata come segue:

```
SET json_serialization_enable TO true;
SET json_serialization_parse_nested_strings TO true;
SELECT name FROM spectrum.customers order by id LIMIT 1;

name
---------
{"given": {"first":"John","middle":"James"}, "family": "Smith"}
```

Anziché essere sottoposto a escape in questo modo:

```
SET json_serialization_enable TO true;
SELECT name FROM spectrum.customers order by id LIMIT 1;

name
---------
{"given": "{\"first\":\"John\",\"middle\":\"James\"}", "family": "Smith"}
```

Utilizzo degli HyperLogLog schizzi in Amazon Redshift

HyperLogLog è un algoritmo utilizzato per stimare la cardinalità di un multiset. Per cardinalità si intende il numero di valori distinti in un multiset. Ad esempio, nell'insieme di {4,3,6,2,2,6,4,3,6,2,2,3}, la cardinalità è 4 con valori distinti di 4, 3, 6 e 2.

La precisione dell' HyperLogLog algoritmo (nota anche come valore m) può influire sulla precisione della cardinalità stimata. Durante la stima della cardinalità, Amazon Redshift utilizza un valore di precisione di default pari a 15. Questo valore può essere fino a 26 per i set di dati più piccoli. Pertanto, l'errore relativo medio varia tra lo 0,01 e lo 0,6%.

Quando si calcola la cardinalità di un multiset, l' HyperLogLog algoritmo genera un costrutto chiamato sketch HLL. Uno schizzo HLL incapsula le informazioni sui valori distinti in un multiset. Il tipo di dati Amazon Redshift HLLSKETCH rappresenta tali valori di schizzo. Questo tipo di dati può essere utilizzato per memorizzare gli schizzi in una tabella Amazon Redshift. Inoltre, Amazon Redshift supporta operazioni che possono essere applicate ai valori HLLSKETCH come funzioni di aggregazione e scalari. È possibile utilizzare queste funzioni per estrarre la cardinalità di un HLLSKETCH e combinare più valori HLLSKETCH.

Il tipo di dati HLLSKETCH offre notevoli vantaggi in termini di prestazioni di query quando si estrae la cardinalità da set di dati di grandi dimensioni. È possibile preaggregare questi set di dati utilizzando i valori HLLSKETCH e memorizzarli nelle tabelle. Amazon Redshift può estrarre la cardinalità direttamente dai valori HLLSKETCH memorizzati senza accedere ai set di dati sottostanti.

Durante l'elaborazione degli schizzi HLL, Amazon Redshift esegue ottimizzazioni che riducono al minimo l'ingombro di memoria dello schizzo e massimizzano la precisione della cardinalità estratta. Amazon Redshift utilizza due rappresentazioni per schizzi HLL, sparse e dense. Un HLLSKETCH inizia in formato sparso. Man mano che vengono inseriti nuovi valori, le sue dimensioni aumentano. Dopo che la sua dimensione raggiunge la dimensione della rappresentazione densa, Amazon Redshift converte automaticamente lo schizzo da sparso a denso.

Amazon Redshift importa, esporta e stampa un HLLSKETCH come JSON quando lo schizzo è in un formato sparso. Amazon Redshift importa, esporta e stampa un HLLSKETCH come stringa Base64 quando lo schizzo è in un formato denso. Per ulteriori informazioni su UNLOAD, consultare Scarico del tipo di dati HLLSKETCH. Per importare dati contenenti testo o valori separati da virgola (CSV) in Amazon Redshift, utilizzare il comando COPY. Per ulteriori informazioni, consulta Caricamento del tipo di dati HLLSKETCH.

Per informazioni sulle funzioni utilizzate con, vedere. HyperLogLog HyperLogLog funzioni

Argomenti

Considerazioni

Di seguito sono riportate le considerazioni relative all'utilizzo HyperLogLog in Amazon Redshift:

- Le seguenti non HyperLogLog funzioni possono accettare un input di tipo HLLSKETCH o colonne di tipo HLLSKETCH:
  - La funzione di aggregazione COUNT
  - Le espressioni condizionali COALESCE e NVL
  - Espressioni CASE
- La codifica supportata è RAW.
- È possibile eseguire un'operazione di UNLOAD sulla tabella con colonne HLLSKETCH in testo o CSV. È possibile usare le colonne UNLOAD HLLSKETCH per scrivere dati HLLSKETCH. Amazon Redshift mostra i dati in un formato JSON per una rappresentazione sparsa o un formato Base64 per una rappresentazione densa. Per ulteriori informazioni su UNLOAD, consultare Scarico del tipo di dati HLLSKETCH.

  Di seguito viene illustrato il formato utilizzato per uno schizzo sparso HyperLogLog rappresentato in formato JSON.

  ```
  {"version":1,"logm":15,"sparse":{"indices":
  [15099259,33107846,37891580,50065963],"values":[2,3,2,1]}}
  ```
- È possibile importare testo o dati CSV in Amazon Redshift utilizzando il comando COPY. Per ulteriori informazioni, consultare Caricamento del tipo di dati HLLSKETCH.
- La codifica di default per HLLSKETCH è RAW. Per ulteriori informazioni, consulta Codifiche di compressione.

Limitazioni

Di seguito sono riportate le limitazioni per l'utilizzo HyperLogLog in Amazon Redshift:

- Le tabelle Amazon Redshift non supportano una colonna HLLSKETH come chiave di ordinamento o chiave di distribuzione per una tabella Amazon Redshift.
- Amazon Redshift non supporta colonne HLLSKETCH nelle clausole ORDER BY, GROUP BY o DISTINCT.
- È possibile scaricare colonne di tipo UNLOAD HLLSKETCH solo in formato testo o CSV. Amazon Redshift scrive quindi i dati HLLSKETCH in un formato JSON o Base64. Per ulteriori informazioni su UNLOAD, consultare UNLOAD.
- Amazon Redshift supporta solo HyperLogLog schizzi con una precisione (valore logm) di 15.
- I driver JDBC e ODBC non supportano il tipo di dati HLLSKETCH. Pertanto, per rappresentare i valori HLLSKETCH il set di risultati utilizza VARCHAR.
- Amazon Redshift Spectrum non supporta nativamente i dati HLLSKETCH. Per questo motivo non è possibile creare o modificare una tabella esterna con una colonna HLLSKETCH.
- I tipi di dati per le funzioni definite dall'utente (UDF) in Python non supportano il tipo di dati HLLSKETCH. Per ulteriori informazioni sulle funzioni definite dall'utente Python, consultare Creazione di una funzione definita dall'utente Python scalare.

Esempi

Esempio: restituzione della cardinalità in una query secondaria

L'esempio seguente restituisce la cardinalità per ogni schizzo in una query secondaria per una tabella denominata Sales.

```
CREATE TABLE Sales (customer VARCHAR, country VARCHAR, amount BIGINT);
INSERT INTO Sales VALUES ('David Joe', 'Greece', 14.5),  ('David Joe', 'Greece',
 19.95), ('John Doe', 'USA', 29.95), ('John Doe', 'USA', 19.95), ('George Spanos',
 'Greece', 9.95), ('George Spanos', 'Greece', 2.95);
```

La query seguente genera uno schizzo HLL per i clienti di ogni paese ed estrae la cardinalità. Questo mostra clienti unici di ogni paese.

```
SELECT hll_cardinality(sketch), country
FROM (SELECT hll_create_sketch(customer) AS sketch, country
        FROM Sales
        GROUP BY country) AS hll_subquery;
```

```
hll_cardinality | country
----------------+---------
            1   | USA
            2   | Greece
 ...
```

Esempio: restituzione di un tipo HLLSKETCH da schizzi combinati in una query secondaria

Nell'esempio seguente viene restituito un singolo tipo HLLSKETCH che rappresenta la combinazione di singoli schizzi da una query secondaria. Gli schizzi vengono combinati utilizzando la funzione di aggregazione HLL_COMBINE.

```
SELECT hll_combine(sketch)
FROM (SELECT hll_create_sketch(customers) AS sketch
        FROM Sales
        GROUP BY country) AS hll_subquery

                                        hll_combine
-----------------------------------------------------------------------------------------
 {"version":1,"logm":15,"sparse":{"indices":[29808639,35021072,47612452],"values":
[1,1,1]}}
(1 row)
```

Esempio: restituisci uno HyperLogLog schizzo combinando più schizzi

Per l'esempio riportato di seguito, si supponga che la tabella `page-users` memorizzi gli schizzi preaggregati per ogni pagina visitata dagli utenti su un determinato sito Web. Ogni riga di questa tabella contiene uno HyperLogLog schizzo che rappresenta tutti gli ID utente che mostrano le pagine visitate.

```
page_users
-- +----------------+-------------+--------------+
-- | _PARTITIONTIME | page        | sketch |
-- +----------------+-------------+--------------+
-- | 2019-07-28     | homepage    | CHAQkAQYA... |
-- | 2019-07-28     | Product A   | CHAQxPnYB... |
-- +----------------+-------------+--------------+
```

Nell'esempio seguente vengono uniti gli schizzi preaggregati e viene generato un singolo schizzo. Questo schizzo incapsula la cardinalità collettiva incapsulata da ogni schizzo.

```
SELECT hll_combine(sketch) as sketch
FROM page_users
```

L'output è simile al seguente.

```
-- +----------------------------------------+
-- | sketch |
-- +----------------------------------------+
-- | CHAQ3sGoCxgCIAuCB4iAIBgTIBgqgIAgAwY.... |
-- +----------------------------------------+
```

Quando viene creato un nuovo schizzo, è possibile utilizzare la funzione HLL_CARDINALITY per ottenere i valori distinti collettivi, come illustrato di seguito.

```
SELECT hll_cardinality(sketch)
FROM (
  SELECT
  hll_combine(sketch) as sketch
  FROM page_users
) AS hll_subquery
```

L'output è simile al seguente.

```
-- +-------+
-- | count |
-- +-------+
-- | 54356 |
-- +-------+
```

Esempio: generazione di HyperLogLog schizzi su dati S3 utilizzando tabelle esterne

Gli esempi seguenti memorizzano nella cache gli HyperLogLog sketch per evitare di accedere direttamente ad Amazon S3 per la stima della cardinalità.

Puoi preaggregare e memorizzare nella cache HyperLogLog gli schizzi in tabelle esterne definite per contenere dati Amazon S3. In questo modo, è possibile estrarre stime di cardinalità senza accedere ai dati di base sottostanti.

Ad esempio, si supponga di aver scaricato un set di file di testo delimitati da tabulazioni in Amazon S3. La query riportata di seguito viene eseguita per definire una tabella esterna denominata `sales` nello schema esterno Amazon Redshift denominato `spectrum`. Il bucket Amazon S3 per questo esempio si trova negli Stati Uniti orientali (Virginia settentrionale). Regione AWS

```
create external table spectrum.sales(
salesid integer,
listid integer,
sellerid smallint,
buyerid smallint,
eventid integer,
dateid integer,
qtysold integer,
pricepaid decimal(8,2),
commission decimal(8,2),
saletime timestamp)
row format delimited
fields terminated by '\t' stored as textfile
location 's3://redshift-downloads/tickit/spectrum/sales/';
```

Si supponga di voler calcolare gli acquirenti distinti che hanno acquistato un articolo in determinate date. A tale scopo, l'esempio seguente genera schizzi per gli ID acquirente per ogni giorno dell'anno e memorizza il risultato nella tabella Amazon Redshift `hll_sales`.

```
CREATE TABLE hll_sales AS
SELECT saletime, hll_create_sketch(buyerid) AS sketch
FROM spectrum.sales
GROUP BY saletime;

SELECT TOP 5 * FROM hll_sales;
```

L'output è simile al seguente.

```
-- hll_sales

-- | saletime        | sketch
    |
```

```
-- +-----------------
+-------------------------------------------------------------------+
-- | 7/22/2008 8:30  | {"version":1,"logm":15,"sparse":{"indices":[9281416],"values":
[1]}}
-- | 2/19/2008 0:38  | {"version":1,"logm":15,"sparse":{"indices":[48735497],"values":
[3]}}
-- | 11/5/2008 4:49  | {"version":1,"logm":15,"sparse":{"indices":[27858661],"values":
[1]}}
-- | 10/27/2008 4:08 | {"version":1,"logm":15,"sparse":{"indices":[65295430],"values":
[2]}}
-- | 2/16/2008 9:37  | {"version":1,"logm":15,"sparse":{"indices":[56869618],"values":
[2]}}
-- +----------------
 +-------------------------------------------------------------------+
```

La seguente query mostra il numero stimato di acquirenti diversi che hanno acquistato un articolo durante il venerdì successivo al Ringraziamento del 2008.

```
SELECT hll_cardinality(hll_combine(sketch)) as distinct_buyers
FROM hll_sales
WHERE trunc(saletime) = '2008-11-28';
```

L'output è simile al seguente.

```
distinct_buyers
---------------
386
```

Si supponga che si desidera il numero di utenti distinti che hanno acquistato un articolo in un determinato intervallo di date. Un esempio potrebbe essere dal venerdì dopo il Ringraziamento al lunedì successivo. Per ottenere questo risultato, la query seguente utilizza la funzione di aggregazione `hll_combine`. Questa funzione consente di evitare il doppio conteggio degli acquirenti che hanno acquistato un articolo in più di un giorno dell'intervallo selezionato.

```
SELECT hll_cardinality(hll_combine(sketch)) as distinct_buyers
FROM hll_sales
WHERE saletime BETWEEN '2008-11-28' AND '2008-12-01';
```

L'output è simile al seguente.

```
distinct_buyers
```

```
---------------
1166
```

Per conservare la `hll_sales` tabella up-to-date, esegui la seguente query alla fine di ogni giornata. In questo modo viene generato uno HyperLogLog schizzo basato sugli ID degli acquirenti che hanno acquistato un articolo oggi e lo aggiunge alla `hll_sales` tabella.

```
INSERT INTO hll_sales
SELECT saletime, hll_create_sketch(buyerid)
FROM spectrum.sales
WHERE TRUNC(saletime) = to_char(GETDATE(), 'YYYY-MM-DD')
GROUP BY saletime;
```

Esecuzione di query sui dati tra database

Grazie alle query tra database in Amazon Redshift, è possibile eseguire query su più database in un cluster Amazon Redshift. Con le query tra database, è possibile eseguire query sui dati da qualsiasi database nel cluster Amazon Redshift, indipendentemente dal database a cui si è connessi. Le query tra database eliminano le copie dei dati e semplificano l'organizzazione dei dati per supportare più gruppi di business dallo stesso data warehouse.

Con le query tra database, è possibile completare le seguenti operazioni:

- Eseguire query sui dati tra i database nel cluster Amazon Redshift.

  Non solo è possibile eseguire query da database a cui si è connessi, ma è anche possibile leggere da qualsiasi altro database per cui si dispone delle autorizzazioni.

  Quando si esegue una query sugli oggetti di database in qualsiasi altro database non connesso, è possibile accedere in lettura solo a tali oggetti del database. È possibile utilizzare le query tra database per accedere ai dati da uno qualsiasi dei database del cluster Amazon Redshift senza doversi connettere a quel database specifico. In questo modo è possibile eseguire query e unire i dati distribuiti su più database nel cluster Amazon Redshift in modo rapido e semplice.

  È inoltre possibile unire set di dati da più database in un'unica query e analizzare i dati utilizzando strumenti di business intelligence (BI) o analisi. Puoi continuare a configurare controlli di accesso granulari a livello di tabella per gli utenti utilizzando i comandi SQL standard di Amazon Redshift. In questo modo, è possibile garantire che gli utenti visualizzino solo i sottoinsiemi pertinenti dei dati per i quali dispongono delle autorizzazioni.
- Query su oggetti.

  È possibile eseguire query su altri oggetti di database utilizzando nomi oggetto completi espressi con la notazione in tre parti. Il percorso completo di qualsiasi oggetto di database è costituito da tre componenti: nome del database, schema e nome dell'oggetto. È possibile accedere a qualsiasi oggetto da qualsiasi altro database utilizzando la notazione del percorso completo, *database_name*.*schema_name*.*object_name*. Per accedere a una colonna particolare, utilizzare *database_name*.*schema_name*.*object_name*.*column_name*.

  È inoltre possibile creare un alias per uno schema in un altro database utilizzando la notazione dello schema esterno. Questo schema esterno fa riferimento a un'altra coppia di database e

schema. La query può accedere all'altro oggetto di database utilizzando la notazione dello schema esterno, *external_schema_name*.*object_name*.

Nella stessa query di sola lettura è possibile eseguire query su vari oggetti di database, ad esempio tabelle utente, viste regolari, viste materializzate e viste di associazione tardiva da altri database.

- Gestione delle autorizzazioni.

  Gli utenti con privilegi di accesso per gli oggetti in qualsiasi database in un cluster Amazon Redshift possono eseguire query su tali oggetti. È possibile concedere privilegi agli utenti e ai gruppi di utenti utilizzando il comando GRANT. È inoltre possibile revocare i privilegi utilizzando il comando REVOKE quando un utente non richiede più l'accesso a oggetti di database specifici.

- Utilizzo dei metadati e degli strumenti di BI.

  È possibile creare uno schema esterno per fare riferimento a uno schema in un altro database Amazon Redshift all'interno dello stesso cluster Amazon Redshift. Per ulteriori informazioni, consultare il comando CREATE EXTERNAL SCHEMA.

  Dopo aver creato i riferimenti allo schema esterno, Amazon Redshift mostra le tabelle sotto lo schema dell'altro database in SVV_EXTERNAL_TABLES e SVV_EXTERNAL_COLUMNS per gli strumenti per esplorare i metadati.

  Per integrare le query tra database con gli strumenti di BI, è possibile utilizzare le seguenti viste di sistema. Queste consentono di visualizzare informazioni sui metadati degli oggetti nei database collegati e in altri database nel cluster Amazon Redshift.

  Di seguito sono riportate le viste di sistema che mostrano tutti gli oggetti Amazon Redshift e gli oggetti esterni di tutti i database nel cluster Amazon Redshift

  - SVV_ALL_COLUMNS
  - SVV_ALL_SCHEMAS
  - SVV_ALL_TABLES

  Di seguito sono riportate le viste di sistema che mostrano tutti gli oggetti Amazon Redshift di tutti i database nel cluster Amazon Redshift

  - SVV_REDSHIFT_COLUMNS
  - SVV_REDSHIFT_DATABASE
  - SVV_REDSHIFT_FUNCTIONS
  - SVV_REDSHIFT_SCHEMAS

- SVV_REDSHIFT_TABLES

Argomenti

Considerazioni

Quando si utilizza la funzione di query tra database in Amazon Redshift, considerare quanto segue:

- Amazon Redshift supporta query tra database sui tipi di nodi ra3.4xlarge, ra3.16xlarge e ra3.xlplus.
- Amazon Redshift supporta l'unione di dati da tabelle o viste in uno o più database nello stesso cluster Amazon Redshift.
- Amazon Redshift Serverless supporta le stesse funzionalità tra database dei cluster Amazon Redshift, in modo da poter unire dati da tabelle o viste in uno o più database in uno spazio dei nomi serverless.
- Tutte le query in una transazione sul database connesso leggono i dati nello stesso stato dell'altro database in cui si trovavano i dati all'inizio della transazione. Questo approccio consente di fornire coerenza transazionale delle query tra i database. Amazon Redshift supporta la coerenza transazionale per le query tra database.
- Per ottenere metadati tra i database, utilizzare le viste di metadati SVV_ALL* e SVV_REDSHIFT*. Non è possibile utilizzare la notazione in tre parti o gli schemi esterni per interrogare tabelle o viste di metadati tra database in information_schema e pg_catalog.

Limitazioni

Quando si utilizza la funzione di query tra database in Amazon Redshift, tenere in considerazione le seguenti limitazioni:

- Quando si esegue una query sugli oggetti di database in qualsiasi altro database non connesso, è possibile accedere in lettura solo a tali oggetti del database.
- Non è possibile eseguire query sulle viste create su altri database che fanno riferimento a oggetti di un altro database.

- È possibile creare viste materializzate e ad associazione tardiva solo su oggetti di altri database nel cluster. Non è possibile creare viste regolari su oggetti di altri database nel cluster.
- Amazon Redshift non supporta le tabelle con privilegi a livello di colonna per le query tra database.
- Amazon Redshift non supporta oggetti di catalogo di query su database AWS Glue federati. Per eseguire query su questi oggetti, creare innanzitutto schemi esterni che fanno riferimento a tali origini dati esterne in ogni database.
- L'esecuzione di query tra database su tabelle con chiavi di ordinamento interlacciate non è supportata.

Esempi di utilizzo di una query tra database

Utilizzare gli esempi seguenti per saperne di più su come configurare una query tra database che faccia riferimento a un database Amazon Redshift.

Per iniziare, creare i database `db1` e `db2` e gli utenti `user1` e `user2` nel cluster Amazon Redshift. Per ulteriori informazioni, consultare CREATE DATABASE e CREA UTENTE.

```
--As user1 on db1
CREATE DATABASE db1;

CREATE DATABASE db2;

CREATE USER user1 PASSWORD 'Redshift01';

CREATE USER user2 PASSWORD 'Redshift01';
```

Come `user1` in `db1`, creare una tabella, concedere i privilegi di accesso a `user2` e inserire i valori in `table1`. Per ulteriori informazioni, consultare GRANT e INSERT.

```
--As user1 on db1
CREATE TABLE table1 (c1 int, c2 int, c3 int);

GRANT SELECT ON table1 TO user2;

INSERT INTO table1 VALUES (1,2,3),(4,5,6),(7,8,9);
```

Come `user2` in `db2`, eseguire una query tra database in `db2`utilizzando la notazione in tre parti.

```
--As user2 on db2
```

```
SELECT * from db1.public.table1 ORDER BY c1;
c1 | c2  | c3
---+-----+----
1  |  2  | 3
4  |  5  | 6
7  |  8  | 9
(3 rows)
```

Come `user2` in `db2`, creare uno schema esterno ed eseguire una query tra database in `db2` utilizzando la notazione dello schema esterno.

```
--As user2 on db2
CREATE EXTERNAL SCHEMA db1_public_sch
FROM REDSHIFT DATABASE 'db1' SCHEMA 'public';

SELECT * FROM db1_public_sch.table1 ORDER BY c1;

c1  | c2 | c3
----+----+----
1   | 2  | 3
4   | 5  | 6
7   | 8  | 9
(3 rows)
```

Per creare viste diverse e concedere autorizzazioni a tali viste, come `user1` in `db1`, completare le seguenti operazioni.

```
--As user1 on db1
CREATE VIEW regular_view AS SELECT c1 FROM table1;

GRANT SELECT ON regular_view TO user2;


CREATE MATERIALIZED VIEW mat_view AS SELECT c2 FROM table1;

GRANT SELECT ON mat_view TO user2;


CREATE VIEW late_bind_view AS SELECT c3 FROM public.table1 WITH NO SCHEMA BINDING;

GRANT SELECT ON late_bind_view TO user2;
```

Come user2 in db2, emettere la seguente query tra database utilizzando la notazione in tre parti per visualizzare la determinata vista.

```
--As user2 on db2
SELECT * FROM db1.public.regular_view;
c1
----
1
4
7
(3 rows)

SELECT * FROM db1.public.mat_view;
c2
----
8
5
2
(3 rows)

SELECT * FROM db1.public.late_bind_view;
c3
----
3
6
9
(3 rows)
```

Come user2 in db2, emettere la seguente query tra database utilizzando la notazione dello schema esterno per eseguire una query sulla vista ad associazione tardiva.

```
--As user2 on db2
SELECT * FROM db1_public_sch.late_bind_view;
c3
----
3
6
9
(3 rows)
```

Come user2 in db2, emettere il seguente comando utilizzando le tabelle connesse in una singola query.

```
--As user2 on db2
CREATE TABLE table1 (a int, b int, c int);

INSERT INTO table1 VALUES (1,2,3), (4,5,6), (7,8,9);

SELECT a AS col_1, (db1.public.table1.c2 + b) AS sum_col2, (db1.public.table1.c3 + c)
 AS sum_col3 FROM db1.public.table1, table1 WHERE db1.public.table1.c1 = a;
col_1 | sum_col2 | sum_col3
------+----------+----------
1     | 4        | 6
4     | 10       | 12
7     | 16       | 18
(3 rows)
```

Nell'esempio seguente sono elencati tutti i database del cluster.

```
select database_name, database_owner, database_type
from svv_redshift_databases
where database_name in ('db1', 'db2');

 database_name | database_owner | database_type
---------------+----------------+---------------
 db1           |            100 | local
 db2           |            100 | local
(2 rows)
```

Nell'esempio seguente sono elencati tutti gli schemi Amazon Redshift di tutti i database del cluster.

```
select database_name, schema_name, schema_owner, schema_type
from svv_redshift_schemas
where database_name in ('db1', 'db2');

 database_name |    schema_name     | schema_owner | schema_type
---------------+--------------------+--------------+-------------
 db1           | pg_catalog         |            1 | local
 db1           | public             |            1 | local
 db1           | information_schema |            1 | local
 db2           | pg_catalog         |            1 | local
 db2           | public             |            1 | local
 db2           | information_schema |            1 | local
(6 rows)
```

Nell'esempio seguente sono elencate tutte le tabelle o le viste Amazon Redshift di tutti i database del cluster.

```
select database_name, schema_name, table_name, table_type
from svv_redshift_tables
where database_name in ('db1', 'db2') and schema_name in ('public');

 database_name | schema_name |     table_name      | table_type
---------------+-------------+---------------------+------------
 db1           | public      | late_bind_view      | VIEW
 db1           | public      | mat_view            | VIEW
 db1           | public      | mv_tbl__mat_view__0 | TABLE
 db1           | public      | regular_view        | VIEW
 db1           | public      | table1              | TABLE
 db2           | public      | table2              | TABLE
(6 rows)
```

Nell'esempio seguente sono elencati tutti gli schemi esterni e di Amazon Redshift di tutti i database del cluster.

```
select database_name, schema_name, schema_owner, schema_type
from svv_all_schemas where database_name in ('db1', 'db2') ;

 database_name |    schema_name     | schema_owner | schema_type
---------------+--------------------+--------------+-------------
 db1           | pg_catalog         |            1 | local
 db1           | public             |            1 | local
 db1           | information_schema |            1 | local
 db2           | pg_catalog         |            1 | local
 db2           | public             |            1 | local
 db2           | information_schema |            1 | local
 db2           | db1_public_sch     |            1 | external
(7 rows)
```

Nell'esempio seguente sono elencati tutte le tabelle esterne e di Amazon Redshift di tutti i database del cluster.

```
select database_name, schema_name, table_name, table_type
from svv_all_tables
where database_name in ('db1', 'db2') and schema_name in ('public');

 database_name | schema_name |     table_name      | table_type
```

```
---------------+-------------+---------------------+------------
 db1           | public      | regular_view        | VIEW
 db1           | public      | mv_tbl__mat_view__0 | TABLE
 db1           | public      | mat_view            | VIEW
 db1           | public      | late_bind_view      | VIEW
 db1           | public      | table1              | TABLE
 db2           | public      | table2              | TABLE
(6 rows)
```

Utilizzo di query tra database con l'editor di query

È possibile utilizzare le query tra database per accedere ai dati da uno qualsiasi dei database del cluster Amazon Redshift senza doversi connettere a quel database specifico. Quando si eseguono query tra database su altri database non connessi, si dispone dell'accesso in lettura solo a tali oggetti di database.

È possibile eseguire query su altri oggetti di database utilizzando nomi di oggetti completi espressi con la notazione in tre parti. Il percorso completo di qualsiasi oggetto di database è costituito da tre componenti: nome del database, schema e nome dell'oggetto. Un esempio è *database_name.schema_name.object_name*.

Per utilizzare query tra database con l'editor di query v2

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Crea un cluster per utilizzare le query tra database nell'Editor di query Amazon Redshift v2. Per ulteriori informazioni, consulta Creazione di un cluster nella Guida alla gestione di Amazon Redshift.
3. Abilitare l'accesso all'editor di query con le autorizzazioni appropriate. Per ulteriori informazioni, consulta Esecuzione di query in un database con l'editor di query v2 nella Guida alla gestione di Amazon Redshift.
4. Dal menu di navigazione, scegli Editor di query v2, quindi collegati a un database nel cluster.

   Quando ci si connette all'editor di query v2 per la prima volta, Amazon Redshift mostra le risorse del database connesso per impostazione predefinita.
5. Scegliere gli altri database a cui si ha accesso per visualizzare gli oggetti di database per questi altri database. Per visualizzare gli oggetti, assicurarsi di disporre delle autorizzazioni appropriate. Dopo aver scelto un database, Amazon Redshift mostra l'elenco degli schemi del database.

Selezionare uno schema per visualizzare l'elenco degli oggetti di database all'interno dello schema.

> **Note**
>
> Amazon Redshift non supporta direttamente gli oggetti del catalogo di query che fanno parte di AWS Glue o di database federati. Per eseguire una query su questi oggetti, creare innanzitutto schemi esterni che fanno riferimento a tali origini dati esterne in ogni database.
> Le query tra database di Amazon Redshift che utilizzano la notazione in tre parti non supportano le tabelle di metadati negli schemi `information_schema` e `pg_catalog`, perché queste viste dei metadati sono specifiche di un database.

6. (Facoltativo) Filtrare l'elenco di tabelle o viste per lo schema selezionato.

Condivisione dei dati in Amazon Redshift

Con la condivisione dei dati di Amazon Redshift, puoi condividere in modo sicuro l'accesso ai dati in tempo reale tra cluster Amazon Redshift Account AWS, gruppi di lavoro e Regioni AWS senza spostare o copiare manualmente i dati. Poiché i dati sono attivi, tutti gli utenti possono visualizzare le informazioni più consistenti up-to-date e coerenti in Amazon Redshift non appena vengono aggiornati.

Puoi condividere dati tra cluster predisposti, gruppi di lavoro serverless, zone di disponibilità e. Account AWS Regioni AWSÈ possibile condividere tra tipi di cluster e tra cluster con provisioning e serverless.

Scritture su più magazzini in Amazon Redshift (anteprima)

Puoi condividere oggetti di database per le letture e le scritture tra diversi cluster Amazon Redshift o gruppi di lavoro Serverless Amazon Redshift all'interno dello Account AWS stesso o da uno all'altro. Account AWS Puoi scrivere dati anche tra le regioni. È possibile assegnare le autorizzazioni come SELECT, INSERT e UPDATE per tabelle diverse e USAGE e CREATE per schemi diversi. I dati sono attivi e disponibili per tutti i warehouse non appena viene eseguita una transazione di scrittura.

Per ulteriori informazioni sulla configurazione delle funzionalità per la condivisione dei dati nella traccia PREVIEW_2023, consulta Condivisione dell'accesso alla scrittura ai dati (anteprima).

> **Note**
>
> Le operazioni di scrittura in più warehouse tramite condivisione dei dati non sono attualmente disponibili nei cluster ra3.xlplus. Per utilizzare questa funzionalità, crea cluster ra3.4xl, cluster ra3.16xl o gruppi di lavoro Serverless Amazon Redshift.

Panoramica della condivisione dei dati in Amazon Redshift

Con la condivisione dei dati, puoi condividere in modo sicuro e semplice dati in tempo reale tra cluster Amazon Redshift.

Per informazioni su come iniziare a lavorare con la condivisione dei dati e gestire le condivisioni di dati utilizzando il, consulta. AWS Management ConsoleGestione delle attività di condivisione dati

Casi d'uso per la condivisione dei dati per Amazon Redshift

La condivisione dei dati di Amazon Redshift è particolarmente utile per questi casi d'uso:

- Supporto dei diversi tipi di carichi di lavoro business-critical: utilizzare un cluster centrale di estrazione, trasformazione e caricamento (ETL) che condivide i dati con più cluster di business intelligence (BI) o di analisi. Questo approccio fornisce isolamento del carico di lavoro in lettura e storno di addebito per singoli carichi di lavoro. È possibile ridimensionare e scalare il singolo calcolo del carico di lavoro in base ai requisiti di prezzo e prestazioni specifici del carico di lavoro.
- Abilitazione della collaborazione tra gruppi: abilitare la collaborazione continua tra team e gruppi di business per analisi più ampie, data science e analisi di impatto tra più prodotti.
- Fornire dati come servizio: condividere i dati come servizio nell'intera organizzazione.
- Condivisione dei dati tra ambienti: condividere i dati tra ambienti di sviluppo, test e produzione. È possibile migliorare l'agilità del team condividendo i dati a diversi livelli di granularità.
- Licenza di accesso ai dati in Amazon Redshift: elenca i set di dati Amazon Redshift nel catalogo che AWS Data Exchange i clienti possono trovare, sottoscrivere e richiedere in pochi minuti.

Casi d'uso di condivisione dei dati e accesso in scrittura (anteprima)

La condivisione dei dati per le scritture ha diversi casi d'uso importanti:

- Aggiorna i dati di origine aziendali relativi al produttore: puoi condividere i dati come servizio all'interno dell'organizzazione, ma in questo modo i consumatori possono anche eseguire azioni sui dati di origine. Ad esempio, possono restituire up-to-date valori o confermare la ricezione dei dati. Questi sono solo un paio di possibili casi d'uso aziendali.
- Inserisci record aggiuntivi sul produttore: i consumatori possono aggiungere record ai dati di origine originali. Questi possono essere contrassegnati come provenienti dal consumatore, se necessario.

Per informazioni specifiche su come eseguire operazioni di scrittura su un datashare, vedere Condivisione dell'accesso in scrittura ai dati (anteprima).

Condivisione di dati a diversi livelli in Amazon Redshift

Con Amazon Redshift, è possibile condividere i dati a diversi livelli. Questi livelli includono database, schemi, tabelle, viste (incluse viste regolari, tardive e materializzate) e funzioni definite dall'utente (FDU) SQL. È possibile creare più unità di condivisione dati per un dato database. Una unità di

condivisione dati può contenere oggetti provenienti da più schemi nel database in cui viene creata la condivisione.

Grazie a questa flessibilità nella condivisione dei dati, si ottiene un controllo degli accessi a grana fine. È possibile personalizzare questo controllo per diversi utenti e aziende che hanno bisogno di accedere ai dati di Amazon Redshift.

Gestione della consistenza dei dati in Amazon Redshift

Amazon Redshift offre coerenza transazionale su tutti i cluster di produttori e consumatori up-to-date e condivide visualizzazioni coerenti dei dati con tutti i consumatori.

È possibile aggiornare continuamente i dati nel cluster producer. Tutte le query su un cluster di consumer all'interno di una transazione leggono lo stesso stato dei dati condivisi. Amazon Redshift non prende in considerazione i dati che sono stati modificati da un'altra transazione nel cluster di producer di cui è stato eseguito il commit dopo l'inizio della transazione sul cluster di consumer. Dopo il commit della modifica dei dati nel cluster producer, le nuove transazioni nel cluster consumer possono eseguire immediatamente una query sui dati aggiornati.

L'elevata coerenza elimina i rischi di report aziendali a bassa affidabilità che potrebbero contenere risultati non validi durante la condivisione dei dati. Questo fattore è particolarmente importante per l'analisi finanziaria o dove i risultati potrebbero essere utilizzati per preparare set di dati che vengono utilizzati per addestrare modelli di machine learning.

Considerazioni sull'utilizzo della condivisione dei dati in Amazon Redshift

Di seguito sono riportate le considerazioni sull'utilizzo delle unità di condivisione dati di Amazon Redshift: Per informazioni sulle limitazioni delle unità di condivisione dati, consulta Limitazioni per la condivisione di dati.

- La condivisione dei dati tra regioni include costi aggiuntivi per il trasferimento di dati tra regioni. Questi costi per il trasferimento dei dati non si applicano all'interno della stessa regione, ma solo tra aree geografiche. Per ulteriori informazioni, consulta Gestione del controllo dei costi per la condivisione dei dati tra regioni.
- Gli utenti dell'unità di condivisione dati continuano a connettersi solo al database cluster locale. Non è possibile connettersi ai database creati da una unità di condivisione dati ma è possibile leggere da tali database.

- Al consumer vengono addebitati tutti i costi di calcolo e di trasferimento di dati tra regioni necessari per eseguire query sui dati del producer. Al producer vengono addebitati i costi per l'archiviazione sottostante dei dati nel cluster con provisioning o nello spazio dei nomi serverless.
- Le prestazioni delle query sui dati condivisi dipendono dalla capacità di calcolo dei cluster consumer.

Gestione della crittografia del cluster

Per condividere i dati tra i cluster Account AWS, sia i cluster di produttori che quelli di consumatori devono essere crittografati.

In Amazon Redshift è possibile attivare la crittografia del database per i cluster per proteggere ulteriormente i dati a riposo. Quando si attiva la crittografia per un cluster, i blocchi di dati e i metadati di sistema vengono crittografati per il cluster e i relativi snapshot. È possibile attivare la crittografia quando si avvia il cluster oppure è possibile modificare un cluster non crittografato in modo che utilizzi la crittografia AWS Key Management Service (AWS KMS). Per ulteriori informazioni sulla crittografia del database Amazon Redshift, consulta Crittografia del database Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Per proteggere i dati in transito, tutti i dati vengono crittografati in transito attraverso lo schema di crittografia del cluster producer. Il cluster consumer adotta questo schema di crittografia quando vengono caricati i dati. Il cluster consumer funziona quindi come un normale cluster crittografato. Anche le comunicazioni tra produttore e consumatore vengono crittografate utilizzando uno schema a chiave condivisa. Per ulteriori informazioni sulla crittografia in transito, consultare Crittografia in transito.

Limitazioni per la condivisione di dati

Di seguito sono riportate le limitazioni nell'utilizzo delle unità di condivisione dati in Amazon Redshift:

- La condivisione dei dati è supportata per tutti i tipi di cluster ra3 forniti (ra3.16xlarge, ra3.4xlarge e ra3.xlplus) e Amazon Redshift Serverless. Non è supportata per altri tipi di cluster.
- Per la condivisione dei dati tra account e regioni, sia il cluster producer che quello consumer e gli spazi dei nomi serverless devono essere crittografati. Questo è per motivi di sicurezza. Tuttavia, non è necessario che condividano la stessa chiave di crittografia.
- È possibile condividere le UDF SQL solo tramite le unità di condivisione dati. Le funzioni Python e Lambda definite dall'utente non sono supportate.

- Se il database del produttore dispone di regole di confronto specifiche, utilizzare le stesse impostazioni di confronto per il database del consumatore.
- Amazon Redshift non supporta l'aggiunta di schemi esterni, tabelle o viste ad associazione tardiva nelle tabelle esterne alle unità di condivisione dati.
- Amazon Redshift non supporta funzioni SQL definite dall'utente nidificate sui cluster di producer.
- Amazon Redshift non supporta la condivisione di tabelle con chiavi di ordinamento interlacciate e viste che fanno riferimento a tabelle con chiavi di ordinamento interlacciate.
- I consumer non possono aggiungere oggetti di unità di condivisione dati a un'altra unità di condivisione dati. I consumer, inoltre, non possono aggiungere viste che fanno riferimento a oggetti di unità di condivisione dati a un'altra unità di condivisione dati.
- Amazon Redshift non supporta l'accesso a un oggetto unità di condivisione dati in cui si è verificato un DDL simultaneo tra le fasi Prepare ed Execute dell'accesso.
- Amazon Redshift non supporta la condivisione di procedure archiviate tramite unità di condivisione dati.
- Amazon Redshift non supporta la condivisione di metadati, viste di sistema e tabelle di sistema.

Regioni in cui è disponibile la condivisione dei dati

La tabella seguente elenca la disponibilità per le funzionalità di condivisione dei dati.

Regione	Condivisione dei dati nella stessa area	Condivisione dei dati tra regioni	AWS Lake Formation condivisioni di dati regolate
Stati Uniti orientali (Virginia settentrionale) (us-east-1)	Sì	Sì	Sì
Stati Uniti orientali (Ohio) (us-east-2)	Sì	Sì	Sì
Stati Uniti occidentali (California settentrionale) (us-west-1)	Sì	Sì	Sì

Regione	Condivisione dei dati nella stessa area	Condivisione dei dati tra regioni	AWS Lake Formation condivisioni di dati regolate
Stati Uniti occidentali (Oregon) (us-west-2)	Sì	Sì	Sì
Asia Pacifico (Mumbai) (ap-south-1)	Sì	Sì	Sì
Asia Pacifico (Hyderabad) (ap-south-2)	Sì	No	No
Asia Pacifico (Tokyo) (ap-northeast-1)	Sì	Sì	Sì
Asia Pacifico (Singapore) (ap-southeast-1)	Sì	Sì	Sì
Asia Pacifico (Sydney) (ap-southeast-2)	Sì	Sì	Sì
Asia Pacifico (Giacarta); (ap-southeast-3)	Sì	No	No
Asia Pacifico (Melbourne) (ap-southeast-4)	Sì	No	No
Asia Pacifico (Seoul) (ap-northeast-2)	Sì	Sì	Sì
Asia Pacifico (Osaka-Locale) (ap-northeast-3)	Sì	No	No

Regione	Condivisione dei dati nella stessa area	Condivisione dei dati tra regioni	AWS Lake Formation condivisioni di dati regolate
Africa (Città del Capo) (af-south-1)	Sì	Sì	No
Canada occidentale (Calgary) (ca-west-1)	Sì	No	No
Canada (Centrale) (ca-central-1)	Sì	Sì	Sì
Europa (Francoforte) (eu-central-1)	Sì	Sì	Sì
Europa (Zurigo) (eu-central-2)	Sì	No	No
Europa (Irlanda) (eu-west-1)	Sì	Sì	Sì
Europa (Londra) (eu-west-2)	Sì	Sì	Sì
Europe (Parigi) (eu-west-3)	Sì	Sì	Sì
Europa (Milano) (eu-south-1)	Sì	No	No
Europa (Spagna) (eu-south-2)	Sì	No	No
Europa (Stoccolma) (eu-north-1)	Sì	Sì	Sì
Medio Oriente (EAU) (me-central-1)	Sì	No	No

Regione	Condivisione dei dati nella stessa area	Condivisione dei dati tra regioni	AWS Lake Formation condivisioni di dati regolate
Medio Oriente (Bahrein) (me-south-1)	Sì	No	No
Israele (Tel Aviv) (il-central-1)	Sì	No	No
Sud America (San Paolo) (sa-east-1)	Sì	Sì	Sì
AWS GovCloud (Stati Uniti orientali) (us-gov-east-1)	Sì	No	Sì
AWS GovCloud (Stati Uniti occidentali) (us-gov-west-1)	Sì	No	Sì

Disponibilità regionale per scritture su più magazzini per la condivisione dei dati

Nel brano PREVIEW_2023, la condivisione dei dati consente operazioni di scrittura e funzionalità di condivisione più granulari. Per ulteriori informazioni su come configurarle, consulta Condivisione dell'accesso in scrittura ai dati (anteprima). Per informazioni sulle aree in cui sono disponibili funzionalità di anteprima, consulta Regioni in cui è disponibile la condivisione dei dati (anteprima).

Che cos'è una unità di condivisione dati?

Una unità di condivisione dati è l'unità di condivisione dei dati in Amazon Redshift. Usa le condivisioni di dati per condividere dati uguali Account AWS o diversi. Account AWS Inoltre, è possibile condividere i dati a scopo di lettura tra diversi cluster Amazon Redshift.

Ogni unità di condivisione dati è associata a un database specifico nel cluster Amazon Redshift.

Un amministratore del cluster producer può anche creare unità di condivisione dati per condividere i dati con altri cluster, denominati condivisioni in uscita. Un amministratore del cluster consumer

può ricevere unità di condivisione dati da altri cluster, denominate come condivisioni in entrata. Per dettagli su producer e consumer, consulta Producer e consumer delle unità di condivisione dati.

Gli oggetti delle unità di condivisione dati sono oggetti di database specifici in un cluster che gli amministratori del cluster producer possono aggiungere alle unità di condivisione dati da condividere con i consumer di dati. Gli oggetti delle unità di condivisione dati sono di sola lettura per i consumer di dati. Esempi di oggetti di unità di condivisione dati sono tabelle, viste e funzioni definite dall'utente. È possibile aggiungere oggetti di unità di condivisione dati alle unità di condivisione dati durante la creazione o la modifica di una unità in qualsiasi momento.

La condivisione dei dati continua a funzionare quando i cluster vengono ridimensionati o il cluster producer viene sospeso.

Esistono diversi tipi di unità di condivisione dati.

Argomenti

Unità di condivisione dati standard

Con le condivisioni di dati standard, puoi condividere i dati tra cluster con provisioning, gruppi di lavoro serverless, zone di disponibilità e. Account AWS Regioni AWSÈ possibile condividere tra tipi di cluster e tra cluster con provisioning e Amazon Redshift serverless.

Per condividere i dati, prendi nota dei seguenti cluster, namespace serverless e identificatori predisposti: Account AWS

- Gli spazi dei nomi dei cluster con provisioning sono identificatori che identificano i cluster con provisioning di Amazon Redshift. Un identificatore univoco globale (GUID) dello spazio dei nomi viene creato automaticamente durante la creazione del cluster con provisioning e collegato al cluster. Un nome della risorsa Amazon (ARN) dello spazio dei nomi è nel formato arn:{partition}:redshift:{region}:{account-id}:namespace:{namespace-guid}. È possibile visualizzare lo spazio dei nomi di un cluster con provisioning nella pagina dei dettagli del cluster nella console Amazon Redshift.

Nel flusso di lavoro di condivisione dei dati, il valore GUID e il cluster spazio dei nomi vengono utilizzati per condividere i dati con i cluster in Account AWS. È inoltre possibile trovare lo spazio dei nomi per il cluster corrente utilizzando la funzione `current_namespace`.

- Gli Spazi dei nomi serverless sono identificatori che identificano Amazon Redshift Serverless. Un identificatore univoco globale (GUID) dello spazio dei nomi viene creato automaticamente durante la creazione di Amazon Redshift Serverless e collegato all'istanza. Un nome della risorsa Amazon (ARN) dello spazio dei nomi serverless è nel formato arn:{partition}:redshift-serverless:{region}:{account-id}:namespace/{namespace-guid}.
- Account AWS possono essere consumatori per le condivisioni di dati e sono rappresentati ciascuno da un ID a 12 cifre. Account AWS

Per le unità di condivisione dati standard, considera quanto segue:

- Quando un cluster producer viene eliminato, Amazon Redshift elimina le unità di condivisione dati create dal cluster producer. Quando si esegue il backup e il ripristino di un cluster producer, le unità di condivisione dati create saranno ancora presenti nel cluster ripristinato. Tuttavia, le autorizzazioni dell'unità di condivisione dati concesse ad altri cluster non sono più valide nel cluster ripristinato. Concedere nuovamente le autorizzazioni di utilizzo delle unità di condivisione dati ai cluster consumer desiderati. Il database consumer sul cluster consumer punta all'unità di condivisione dati dal cluster originale in cui viene preso lo snapshot. Per eseguire una query sui dati condivisi dal cluster ripristinato, l'amministratore del cluster consumer crea un database diverso. In alternativa, l'amministratore può eliminare e ricreare un database consumer esistente per utilizzare l'unità condivisione dati dal cluster appena ripristinato.
- Quando un cluster consumer viene eliminato e ripristinato da uno snapshot, l'accesso condiviso precedente a questo cluster non sarà più valido e visibile. Se l'accesso alle unità di condivisione dati è ancora necessario nel cluster consumer ripristinato, l'amministratore del cluster producer deve concedere nuovamente l'utilizzo delle unità di condivisione dati al cluster consumer ripristinato. L'amministratore del cluster consumer deve eliminare tutti i database consumer obsoleti creati dalle unità di condivisione dati inattive. Quindi l'amministratore deve ricreare il database consumer dall'unità di condivisione dati, dopo che il produttore ha nuovamente concesso le autorizzazioni. Poiché il GUID dello spazio dei nomi del cluster è diverso in un cluster ripristinato dal cluster originale, concedere nuovamente le autorizzazioni dell'unità di condivisione dati quando il cluster consumer o producer viene ripristinato dal backup.

AWS Data Exchange condivisioni di dati

Un AWS Data Exchange datashare è un'unità di licenza tramite cui condividere i dati. AWS Data Exchange AWS gestisce tutta la fatturazione e i pagamenti associati agli abbonamenti AWS Data Exchange e all'uso della condivisione dei dati di Amazon Redshift. I fornitori di dati approvati possono aggiungere condivisioni di AWS Data Exchange dati ai prodotti. AWS Data Exchange Quando i clienti si abbonano a un prodotto con AWS Data Exchange datashare, ottengono l'accesso alle condivisioni di dati del prodotto.

AWS Data Exchange per Amazon Redshift semplifica l'accesso in licenza ai dati di Amazon Redshift tramite. AWS Data Exchange Quando un cliente si abbona a un prodotto con condivisioni di AWS Data Exchange dati, aggiunge AWS Data Exchange automaticamente il cliente come consumatore di dati su tutte le AWS Data Exchange condivisioni di dati incluse nel prodotto. Le fatture vengono generate automaticamente e i pagamenti vengono raccolti centralmente ed erogati automaticamente. AWS Marketplace Entitlement Service

I provider possono concedere in licenza i dati in Amazon Redshift a livello granulare, come schemi, tabelle, viste e funzioni definite dall'utente. Puoi utilizzare lo stesso AWS Data Exchange datashare su più prodotti. AWS Data Exchange Tutti gli oggetti aggiunti al AWS Data Exchange datashare sono disponibili per i consumatori. I produttori possono visualizzare tutte le AWS Data Exchange condivisioni di dati gestite da per loro AWS Data Exchange conto utilizzando le operazioni API di Amazon Redshift, i comandi SQL e la console Amazon Redshift. I clienti che sottoscrivono le condivisioni di AWS Data Exchange dati di un prodotto hanno accesso in sola lettura agli oggetti contenuti nelle condivisioni di dati.

I clienti che desiderano utilizzare i dati di produttori di terze parti possono sfogliare il AWS Data Exchange catalogo per scoprire e abbonarsi ai set di dati in Amazon Redshift. Una volta che l' AWS Data Exchange abbonamento è attivo, possono creare un database dal datashare del cluster e interrogare i dati in Amazon Redshift.

Come funzionano le condivisioni di dati AWS Data Exchange

Gestire i AWS Data Exchange datashare in qualità di amministratore del produttore

Se sei un produttore di dati (noto anche come provider on AWS Data Exchange), puoi creare AWS Data Exchange condivisioni di dati che si connettono ai tuoi database Amazon Redshift. Per aggiungere AWS Data Exchange datashare ai prodotti su AWS Data Exchange, devi essere un provider registrato. AWS Data Exchange

Per ulteriori informazioni su come iniziare a utilizzare le AWS Data Exchange condivisioni di dati, consulta. Condivisione di dati con licenza Amazon Redshift su AWS Data Exchange

Utilizzo AWS Data Exchange dei datashare come consumatore con un abbonamento attivo AWS Data Exchange

Se sei un consumatore con un AWS Data Exchange abbonamento attivo (noto anche come abbonato AWS Data Exchange), puoi sfogliare il AWS Data Exchange catalogo sulla AWS Data Exchange console per scoprire i prodotti contenenti datashare. AWS Data Exchange

Dopo esserti abbonato a un prodotto che contiene AWS Data Exchange datashare, crea un database dal datashare all'interno del cluster. È quindi possibile eseguire una query diretta dei dati in Amazon Redshift senza estrarre, trasformare e caricare i dati.

Per ulteriori informazioni su come iniziare a utilizzare i datashare, consulta AWS Data Exchange . Condivisione di dati con licenza Amazon Redshift su AWS Data Exchange

Per le unità di condivisione dati AWS Data Exchange , considera quanto segue:

- Quando un cluster producer viene eliminato, Amazon Redshift elimina le unità di condivisione dati create dal cluster producer. Quando si esegue il backup e il ripristino di un cluster producer, le unità di condivisione dati create saranno ancora presenti nel cluster ripristinato. Per consentire agli abbonati ai dati di continuare ad accedere ai dati, è necessario creare nuovamente le condivisioni di AWS Data Exchange dati e pubblicarle nei set di dati del prodotto. Il database consumer sul cluster consumer punta all'unità di condivisione dati dal cluster originale in cui viene preso lo snapshot. Per interrogare i dati condivisi dal cluster ripristinato, l'amministratore del cluster consumer crea un database diverso oppure elimina e ricrea un database consumer esistente per utilizzare il AWS Data Exchange datashare appena creato dal cluster appena ripristinato.
- Quando un cluster consumer viene eliminato e ripristinato da uno snapshot, l'accesso condiviso precedente a questo cluster non sarà più valido e visibile. L'amministratore del cluster consumer deve eliminare tutti i database consumer obsoleti creati dalle unità di condivisione dati inattive e creare nuovamente il database consumer dall'unità di condivisione dati dopo che il producer ha nuovamente concesso le autorizzazioni. Poiché il GUID dello spazio dei nomi del cluster è diverso in un cluster ripristinato rispetto al cluster originale, concedi nuovamente le autorizzazioni dell'unità di condivisione dati quando il cluster consumer o producer viene ripristinato dal backup.
- Ti consigliamo di non eliminare il cluster se disponi di condivisioni di dati. AWS Data Exchange L'esecuzione di questo tipo di alterazione può violare i termini del prodotto dei dati in AWS Data Exchange.

Considerazioni sull'utilizzo AWS Data Exchange per Amazon Redshift

Quando lo utilizzi AWS Data Exchange per Amazon Redshift, considera quanto segue:

- Sia i produttori che i consumatori devono utilizzare i tipi di istanza RA3 per utilizzare le unità di condivisione dati di Amazon Redshift. I produttori devono utilizzare i tipi di istanza RA3 con la versione più recente del cluster Amazon Redshift.
- Entrambi i cluster producer e consumer devono essere crittografati.
- Devi essere registrato come AWS Data Exchange fornitore per pubblicare offerte di prodotti AWS Data Exchange, compresi i prodotti che contengono condivisioni di AWS Data Exchange dati. Per ulteriori informazioni, consultare Guida introduttiva come provider.
- Non è necessario essere un AWS Data Exchange provider registrato per trovare, abbonarsi e interrogare i dati di Amazon Redshift. AWS Data Exchange
- Per controllare l'accesso ai tuoi dati, crea condivisioni di AWS Data Exchange dati con l'impostazione accessibile al pubblico attivata. Per modificare un AWS Data Exchange datashare in modo da disattivare l'impostazione accessibile al pubblico, imposta la variabile di sessione in modo da consentire ALTER DATASHARE SET PUBLICACCESSIBLE FALSE. Per ulteriori informazioni, consulta Note per l'utilizzo di ALTER DATASHARE.
- I produttori non possono aggiungere o rimuovere manualmente i consumatori dalle condivisioni di dati perché l'accesso ai AWS Data Exchange datashare viene concesso sulla base di un abbonamento attivo a un prodotto che contiene il datashare. AWS Data Exchange AWS Data Exchange
- I produttori non possono visualizzare le query SQL eseguite dai consumatori. Possono visualizzare solo i metadati, ad esempio il numero di query o gli oggetti interrogati dai consumatori, tramite tabelle Amazon Redshift a cui solo il produttore può accedere. Per ulteriori informazioni, consulta Monitoraggio e verifica della condivisione dati in Amazon Redshift.
- Ti consigliamo di rendere accessibili al pubblico le tue unità di condivisione dati. In caso contrario, gli abbonati che utilizzano cluster di consumatori accessibili AWS Data Exchange al pubblico non saranno in grado di utilizzare il datashare.
- Ti consigliamo di non eliminare un AWS Data Exchange datashare condiviso con altri Account AWS utilizzando l'istruzione DROP DATASHARE. In tal caso, coloro Account AWS che hanno accesso al datashare perderanno l'accesso. Questa operazione è irreversibile. L'esecuzione di questo tipo di alterazione può violare i termini del prodotto dei dati in AWS Data Exchange. Se desideri eliminare un AWS Data Exchange datashare, consulta. Note per l'utilizzo di DROP DATASHARE

- Per la condivisione di dati tra regioni, puoi creare AWS Data Exchange datashare per condividere i dati concessi in licenza.
- Quando utilizza dati da una regione diversa, il consumer paga la tariffa per il trasferimento di dati tra regioni dalla regione producer alla regione consumer.

Unità di condivisione dati gestite da AWS Lake Formation

Utilizzando AWS Lake Formation, puoi definire e applicare centralmente le autorizzazioni di accesso a livello di database, tabelle, colonne e righe delle condivisioni di dati Amazon Redshift e limitare l'accesso degli utenti agli oggetti all'interno di un datashare. La condivisione dei dati tramite Lake Formation permette di definire le autorizzazioni in Lake Formation e applicarle a qualsiasi unità di condivisione dati e ai relativi oggetti. Ad esempio, se si dispone di una tabella contenente informazioni sui dipendenti, è possibile utilizzare i filtri a livello di colonna di Lake Formation per impedire ai dipendenti che non lavorano nel reparto delle risorse umane di visualizzare informazioni di identificazione personale (PII), come il numero di previdenza sociale. Per ulteriori informazioni, consulta Filtraggio dei dati e sicurezza a livello di cella in Lake Formation nella Guida per gli sviluppatori di AWS Lake Formation .

È anche possibile usare i tag in Lake Formation per configurare le autorizzazioni sulle risorse di Lake Formation. Per ulteriori informazioni, consulta Controllo degli accessi basato su tag di Lake Formation.

Attualmente Amazon Redshift supporta la condivisione dati tramite Lake Formation all'interno dello stesso account o tra account diversi. La condivisione tra regioni non è al momento supportata.

Di seguito è riportata una panoramica generale su come utilizzare Lake Formation per controllare le autorizzazioni relative alle unità di condivisione dati:

1. In Amazon Redshift, l'amministratore del gruppo di lavoro o del cluster producer crea un'unità di condivisione dati sul gruppo di lavoro o sul cluster producer e ne assegna l'utilizzo a un account Lake Formation.
2. L'amministratore del gruppo di lavoro o del cluster producer autorizza l'account Lake Formation ad accedere all'unità di condivisione dati.
3. L'amministratore di Lake Formation rileva e registra le unità di condivisione dati. Devono inoltre scoprire gli AWS Glue ARN a cui hanno accesso e associare le condivisioni di dati a un ARN. AWS Glue Data Catalog Se utilizzi il, AWS CLI puoi scoprire e accettare le condivisioni di dati con le operazioni della CLI di Redshift e. `describe-data-shares` `associate-`

`data-share-consumer` Per registrare una unità di condivisione dati, utilizza l'operazione CLI `register-resource` di Lake Formation.

4. L'amministratore di Lake Formation crea un database federato in e configura le AWS Glue Data Catalog autorizzazioni di Lake Formation per controllare l'accesso degli utenti agli oggetti all'interno del datashare. Per ulteriori informazioni sui database federati in AWS Glue, consulta Gestione delle autorizzazioni per i dati in un datashare Amazon Redshift.
5. L'amministratore di Lake Formation scopre i AWS Glue database a cui ha accesso e associa il datashare a un ARN. AWS Glue Data Catalog
6. L'amministratore di Redshift scopre gli ARN del AWS Glue database a cui ha accesso, crea un database esterno nel cluster di consumatori Amazon Redshift utilizzando un AWS Glue ARN del database e concede l'utilizzo agli utenti del database autenticati con credenziali IAM per iniziare a interrogare il database Amazon Redshift.
7. Gli utenti del database possono utilizzare le viste SVV_EXTERNAL_TABLES e SVV_EXTERNAL_COLUMNAL_COLUMNS per trovare tutte le tabelle o le colonne all'interno del AWS Glue database a cui hanno accesso, quindi possono interrogare le tabelle del AWS Glue database.
8. Quando l'amministratore del gruppo di lavoro o del cluster producer decide di non condividere più i dati con il cluster consumer, può revocare l'utilizzo, annullare l'autorizzazione o eliminare l'unità di condivisione dati da Redshift. Le autorizzazioni e gli oggetti associati in Lake Formation non vengono eliminati automaticamente.

Per ulteriori informazioni sulla condivisione di un datashare con un produttore, un cluster o un amministratore di un gruppo di lavoro, vedere. AWS Lake Formation Utilizzo di unità di condivisione dati gestite da Lake Formation in qualità di producer Per utilizzare i dati condivisi dal gruppo di lavoro o dal cluster producer, consulta Utilizzo di unità di condivisione dati gestite da Lake Formation in qualità di consumer.

Considerazioni e limitazioni relative all'utilizzo AWS Lake Formation con Amazon Redshift

Di seguito sono riportate alcune considerazioni e limitazioni della condivisione dei dati di Amazon Redshift attraverso Lake Formation. Per informazioni su considerazioni e limitazioni relative alla condivisione dei dati, consulta Considerazioni sulla condivisione dei dati in Amazon Redshift. Per informazioni sulle limitazioni di Lake Formation, consulta Note sull'utilizzo delle unità di condivisione dati Amazon Redshift in Lake Formation.

- Al momento la condivisione tra regioni di un'unità di condivisione dati su Lake Formation non è supportata.
- Se i filtri a livello di colonna sono definiti per un utente in una relazione condivisa, l'esecuzione di un'operazione `SELECT *` restituisce solo le colonne a cui l'utente ha accesso.
- I filtri a livello di cella di Lake Formation non sono supportati.
- Se è stata creata e condivisa una vista e le relative tabelle con Lake Formation, è possibile configurare filtri per gestire l'accesso alle tabelle. Amazon Redshift applica policy definite da Lake Formation quando gli utenti del cluster consumer accedono agli oggetti condivisi. Quando un utente accede a una vista condivisa con Lake Formation, Redshift applica solo le policy di Lake Formation definite nella vista e non le tabelle contenute nella vista. Tuttavia, quando gli utenti accedono direttamente alla tabella, Redshift applica le policy di Lake Formation definite sulla tabella.
- Non puoi creare viste materializzate sul consumer basate su una tabella condivisa se nella tabella sono configurati filtri di Lake Formation.
- L'amministratore di Lake Formation deve disporre delle autorizzazioni di amministratore del data lake e delle autorizzazioni necessarie per accettare una unità di condivisione dati.
- Per condividere le unità di condivisione dati tramite Lake Formation, il cluster producer consumer deve essere un cluster RA3 con la versione più recente del cluster Amazon Redshift o un gruppo di lavoro serverless.
- Entrambi i cluster producer e consumer devono essere crittografati.
- Le policy di controllo degli accessi a livello di riga e colonna di Redshift implementate nel gruppo di lavoro o nel cluster producer vengono ignorate quando l'unità di condivisione dati viene condivisa con Lake Formation. L'amministratore di Lake Formation deve configurare queste policy in Lake Formation. L'amministratore del gruppo di lavoro o del cluster producer può disattivare RLS per una tabella utilizzando il comando ALTER TABLE.
- La condivisione di unità di condivisione dati tramite Lake Formation è disponibile solo per gli utenti che hanno accesso sia a Redshift che a Lake Formation.

Producer e consumer delle unità di condivisione dati

I producer di dati (noti anche come producer di condivisione dati o producer di unità di condivisione dati) sono cluster da cui si desidera condividere i dati. Gli amministratori del cluster produttore e i proprietari di database possono creare unità di condivisione dati utilizzando il comando CREA UNITÁ CONDIVISIONE DATI. Puoi aggiungere oggetti come schemi, tabelle, viste e funzioni SQL definite

dall'utente (UDF) da un database che desideri che il cluster di produttori condivida con i cluster di consumatori.

I produttori di dati (noti anche come provider on AWS Data Exchange) per le AWS Data Exchange condivisioni di dati possono concedere in licenza i dati. AWS Data Exchange I provider approvati possono aggiungere condivisioni di AWS Data Exchange dati ai prodotti. AWS Data Exchange

Quando un cliente si abbona a un prodotto con AWS Data Exchange datashare, aggiunge AWS Data Exchange automaticamente il cliente come consumatore di dati su tutte le AWS Data Exchange condivisioni di dati incluse nel prodotto. AWS Data Exchange rimuove inoltre tutti i clienti dalle condivisioni di AWS Data Exchange dati al termine dell'abbonamento. AWS Data Exchange gestisce inoltre automaticamente la fatturazione, la fatturazione, la riscossione dei pagamenti e la distribuzione dei pagamenti per i prodotti a pagamento con datashare. AWS Data Exchange Per ulteriori informazioni, consulta AWS Data Exchange condivisioni di dati. Per registrarti come provider di dati AWS Data Exchange , consultare Guida introduttiva come provider.

I consumer di dati (noti anche come consumatori di condivisione dei dati o consumer di unità di condivisione dati) sono cluster che ricevono unità di condivisione dati dai cluster producer.

I cluster Amazon Redshift che condividono dati possono essere uguali, diversi Account AWS o diversi Regioni AWS, in modo da poter condividere i dati tra le organizzazioni e collaborare con altre parti. Gli amministratori del cluster consumer ricevono le unità di condivisione dati per le quali è loro concesso l'utilizzo e rivedono i contenuti di ciascuna unità di condivisione dati. Per utilizzare i dati condivisi, l'amministratore del cluster consumer crea un database Amazon Redshift dall'unità di condivisione dati. L'amministratore assegna quindi le autorizzazioni per il database agli utenti e ai ruoli nel cluster consumer. Una volta fornite le autorizzazioni, gli utenti e i ruoli possono elencare gli oggetti condivisi come parte delle query di metadati standard, insieme ai dati locali nel cluster consumer. Possono iniziare a eseguire query immediatamente.

Se sei un consumatore con un AWS Data Exchange abbonamento attivo (noto anche come abbonato su AWS Data Exchange), puoi trovare, sottoscrivere ed eseguire query su up-to-date dati granulari in Amazon Redshift senza la necessità di estrarli, trasformarli e caricarli. Per ulteriori informazioni, consulta AWS Data Exchange condivisioni di dati.

Come funziona la condivisione dei dati in Amazon Redshift

Gestione delle unità di condivisione dati in diversi stati

Con le unità di condivisione dati tra account, esistono diversi stati delle unità che richiedono l'intervento dell'utente. L'unità di condivisione dati può avere lo stato attivo, azione richiesta o inattivo.

Di seguito viene descritto ogni stato dell'unità di condivisione dati e la relativa operazione richiesta:

- Quando un amministratore del cluster producer crea una unità di condivisione dati, lo stato dell'unità di condivisione dati nel cluster producer è In attesa di autorizzazione. L'amministratore del cluster producer può autorizzare i consumer di dati ad accedere all'unità di condivisione dati. L'amministratore del cluster consumer non ha alcuna operazione da eseguire.
- Quando un amministratore del cluster producer autorizza l'unità di condivisione dati, lo stato dell'unità di condivisione dati nel cluster producer diventa Autorizzato. L'amministratore del cluster producer non ha alcuna operazione da eseguire. Quando esiste almeno un'associazione con un consumer di dati per l'unità di condivisione dati, lo stato dell'unità cambia da Autorizzato a Attivo.

  Lo stato di condivisione dell'unità di condivisione diventa quindi Disponibile (azione richiesta sulla console Amazon Redshift) nel cluster consumer. L'amministratore del cluster consumer può associare l'unità di condivisione dati con i consumer di dati oppure rifiutare l'unità. L'amministratore del cluster consumer può anche utilizzare il comando `describeDatashareforConsumer` della AWS CLI per visualizzare lo stato delle unità di condivisione dati. In alternativa, per visualizzare lo stato dell'unità, l'amministratore può usare il comando della CLI `describeDatashare` e fornire l'Amazon Resource Name (ARN) dell'unità di condivisione dati.
- Quando l'amministratore del cluster consumer associa una unità di condivisione dati ai consumer di dati, lo stato dell'unità diventa Attivo nel cluster producer. Se esiste almeno un'associazione con un consumer di dati per l'unità di condivisione dati, lo stato dell'unità cambia da Autorizzato a Attivo. Non è presente alcuna operazione richiesta per l'amministratore del cluster producer.

  Lo stato dell'unità di condivisione dati diventa Attivo nel cluster consumer. Non è presente alcuna operazione richiesta per l'amministratore del cluster consumer.
- Quando l'amministratore del cluster consumer rimuove un'associazione consumer da una unità di condivisione dati, lo stato dell'unità diventa Attivo o Autorizzato. Diventa Attivo se per l'unità di condivisione dati esiste almeno un'associazione con un altro consumer di dati. Diventa Autorizzato quando non esiste alcuna associazione di consumer con l'unità di condivisione dati sul cluster producer. L'amministratore del cluster producer non ha alcuna operazione da eseguire.

Se tutte le associazioni vengono rimosse, lo stato dell'unità di condivisione dati diventa Operazione richiesta nel cluster consumer. Una volta che l'unità di condivisione dati torna disponibile ai consumer, l'amministratore del cluster consumer potrà riassociare l'unità di condivisione dati ai consumer di dati.

- Quando un amministratore di cluster consumer rifiuta una unità di condivisione dati, lo stato dell'unità nel cluster producer diventa Operazione richiesta e Rifiutato nel cluster consumer. L'amministratore del cluster producer può autorizzare nuovamente l'unità di condivisione dati. L'amministratore del cluster consumer non ha alcuna operazione da eseguire.
- Quando l'amministratore del cluster producer rimuove l'autorizzazione da una unità di condivisione dati, lo stato dell'unità nel cluster producer diventa Operazione richiesta. Se necessario, l'amministratore del cluster di producer può decidere di autorizzare nuovamente l'unità di condivisione dati. Non è presente alcuna operazione richiesta per l'amministratore del cluster consumer.

Condivisione di unità di condivisione dati

È necessario utilizzare le unità di condivisione dati solo quando si condividono dati tra diversi cluster Amazon Redshift con provisioning o gruppi di lavoro serverless. All'interno dello stesso cluster, è possibile eseguire query su un altro database utilizzando una semplice notazione in tre parti `database.schema.table` purché si disponga delle autorizzazioni richieste per gli oggetti dell'altro database.

Gestione delle autorizzazioni per le unità di condivisione dati in Amazon Redshift

Un amministratore del cluster di producer mantiene il controllo per i set di dati che condivide. È possibile aggiungere nuovi oggetti o rimuoverli dall'unità di condivisione dati. Puoi anche concedere o revocare l'accesso alle condivisioni di dati nel loro insieme per i cluster di consumatori, gli account o le regioni. AWS AWS Quando le autorizzazioni vengono revocate, i cluster consumer perdono immediatamente l'accesso agli oggetti condivisi e smettono di vederli nell'elenco delle unità di condivisione dati INBOUND in SVV_DATASHARES.

L'esempio seguente crea il datashare`salesshare`, aggiunge lo schema `public` e aggiunge la tabella a. `public.tickit_sales_redshift` `salesshare` Concede inoltre le autorizzazioni di utilizzo per lo spazio dei nomi del `salesshare` cluster specificato.

```
CREATE DATASHARE salesshare;

ALTER DATASHARE salesshare ADD SCHEMA public;

ALTER DATASHARE salesshare ADD TABLE public.tickit_sales_redshift;

GRANT USAGE ON DATASHARE salesshare TO NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

Per CREATE DATASHARE, gli utenti con privilegi avanzati e i proprietari di database possono creare le unità di condivisione dati. Per ulteriori informazioni, consulta CREARE DATASHARE. Per ALTER DATASHARE, il proprietario dell'unità di condivisione dati con le autorizzazioni necessarie per gli oggetti dell'unità di condivisione dati da aggiungere o rimuovere può modificare l'unità. Per informazioni, consultare ALTER DATASHARE.

L'amministratore producer può eliminare una unità di condivisione dati e questa non sarà più visualizzata nei cluster consumer. I database e i riferimenti allo schema creati nel cluster consumer dai dati rilasciati continuano a essere presenti, pure senza contenere oggetti. L'amministratore del cluster di consumer deve eliminare manualmente questi database.

Sul lato consumer, un amministratore del cluster consumer può determinare quali utenti e ruoli devono poter accedere ai dati condivisi creando un database dall'unità di condivisione dati. A seconda delle opzioni scelte durante la creazione del database, è possibile controllare l'accesso nel modo seguente. Per ulteriori informazioni sulla creazione di un database da un'unità di condivisione dati, consulta CREATE DATABASE.

Creazione del database senza la clausola WITH PERMISSIONS

Un amministratore può controllare l'accesso a livello di database o schema. Per controllare l'accesso a livello di schema, l'amministratore deve creare uno schema esterno dal database Amazon Redshift creato dall'unità di condivisione dati.

Nell'esempio seguente vengono concesse le autorizzazioni per accedere a una tabella condivisa a livello di database e di schema.

```
GRANT USAGE ON DATABASE sales_db TO Bob;

CREATE EXTERNAL SCHEMA sales_schema FROM REDSHIFT DATABASE sales_db SCHEMA 'public';

GRANT USAGE ON SCHEMA sales_schema TO ROLE Analyst_role;
```

Per limitare ulteriormente l'accesso, è possibile creare viste sopra gli oggetti condivisi, esponendo solo i dati necessari. È possibile utilizzare queste viste per consentire l'accesso a utenti e ruoli.

Una volta ottenuto l'accesso al database o allo schema, gli utenti potranno accedere a tutti gli oggetti condivisi in quel database o schema.

Creazione del database con la clausola WITH PERMISSIONS

Dopo aver assegnato i diritti di utilizzo per il database o lo schema, un amministratore può controllare ulteriormente l'accesso utilizzando lo stesso processo di assegnazione delle autorizzazioni utilizzato per un database o uno schema locale. Senza le autorizzazioni relative ai singoli oggetti, gli utenti non possono accedere a nessun oggetto nel database o nello schema dell'unità di condivisione dati anche se dispongono dell'autorizzazione USAGE.

L'esempio seguente assegna le autorizzazioni per accedere a una tabella condivisa a livello di database.

```
GRANT USAGE ON DATABASE sales_db TO Bob;
GRANT USAGE FOR SCHEMAS IN DATABASE sales_db TO Bob;
GRANT SELECT ON sales_db.public.tickit_sales_redshift TO Bob;
```

Dopo aver ottenuto l'accesso al database o allo schema, gli utenti devono comunque ricevere le autorizzazioni pertinenti per tutti gli oggetti nel database o nello schema a cui si desidera che accedano.

Condivisione granulare utilizzando WITH PERMISSIONS (anteprima)

Abilitazione dell'esecuzione di query per cluster o gruppi di lavoro serverless sull'unità di condivisione dati

In questo passaggio si presuppone che l'unità di condivisione dati abbia origine da un altro cluster o un altro spazio dei nomi Amazon Redshift serverless del tuo account oppure provenga da un altro account associato allo spazio dei nomi che utilizzi.

1. L'amministratore del database consumer può creare un database dall'unità di condivisione dati.

   ```
   CREATE DATABASE my_ds_db [WITH PERMISSIONS] FROM DATASHARE my_datashare OF
    NAMESPACE 'abc123def';
   ```

Se crei un database con la clausola WITH PERMISSIONS, puoi assegnare autorizzazioni granulari per gli oggetti dell'unità di condivisione dati a utenti e ruoli diversi. In caso contrario, a tutti gli utenti e i ruoli a cui è stata concessa l'autorizzazione USAGE per il database dell'unità di condivisione dati vengono assegnate tutte le autorizzazioni per tutti gli oggetti all'interno del database dell'unità di condivisione dati.

2. Di seguito viene illustrato come assegnare le autorizzazioni a un utente o un ruolo del database Redshift. È necessario essere connessi a un database locale per eseguire queste istruzioni. Non è possibile usare queste istruzioni se si esegue un comando USE sul database dell'unità di condivisione dati prima delle istruzioni grant.

```
GRANT USAGE ON DATABASE my_ds_db TO ROLE data_eng;
GRANT CREATE, USAGE ON SCHEMA my_ds_db.my_shared_schema TO ROLE data_eng;
GRANT ALL ON ALL TABLES IN SCHEMA my_ds_db.my_shared_schema TO ROLE data_eng;

GRANT USAGE ON DATABASE my_ds_db TO bi_user;
GRANT USAGE ON SCHEMA my_ds_db.my_shared_schema TO bi_user;
GRANT SELECT ON my_ds_db.my_shared_schema.table1 TO bi_user;
```

Utilizzo delle viste nella condivisione dei dati Amazon Redshift

Un cluster producer può condividere viste regolari, ad associazione tardiva e materializzate. Quando si condividono viste regolari o ad associazione tardiva, non è necessario condividere le tabelle di base. La tabella seguente mostra come sono supportate le viste con la condivisione dati.

Nome della vista	Questa vista può essere aggiunta a una unità di condivisione dati?	Un consumer può creare questa vista su oggetti dell'unità di condivisione dati tra cluster?	
Vista regolare	Sì	No	
Vista ad associazione tardiva	Sì	Sì	

Nome della vista	Questa vista può essere aggiunta a una unità di condivisione dati?	Un consumer può creare questa vista su oggetti dell'unità di condivisione dati tra cluster?	
Vista materializzata	Sì	Sì, ma solo con un aggiornamento completo	

La query seguente mostra l'output di una vista regolare supportata con la condivisione dei dati. Per ulteriori informazioni sulla definizione di vista regolare, consultare CREATE VIEW.

```
SELECT * FROM tickit_db.public.myevent_regular_vw
ORDER BY eventid LIMIT 5;

   eventid  |  eventname
  ----------+-------------
     3835   | LeAnn Rimes
     3967   | LeAnn Rimes
     4856   | LeAnn Rimes
     4948   | LeAnn Rimes
     5131   | LeAnn Rimes
```

La query seguente mostra l'output di una vista ad associazione tardiva supportata con la condivisione dei dati. Per ulteriori informazioni sulla vista ad associazione tardiva, consultare CREATE VIEW.

```
SELECT * FROM tickit_db.public.event_lbv
ORDER BY eventid LIMIT 5;

 eventid | venueid | catid | dateid |          eventname           |      starttime
 --------+---------+-------+--------+------------------------------
+---------------------
     1   |   305   |   8   |  1851  |        Gotterdammerung       | 2008-01-25
 14:30:00
     2   |   306   |   8   |  2114  |           Boris Godunov      | 2008-10-15
 20:00:00
     3   |   302   |   8   |  1935  |              Salome          | 2008-04-19
 14:30:00
```

```
    4   |   309   |   8   |  2090  |  La Cenerentola (Cinderella) | 2008-09-21
 14:30:00
    5   |   302   |   8   |  1982  |          Il Trovatore        | 2008-06-05
 19:00:00
```

La query seguente mostra l'output di una vista materializzata supportata con la condivisione dei dati. Per ulteriori informazioni sulla definizione di vista materializzata, consultare CREATE MATERIALIZED VIEW.

```
SELECT * FROM tickit_db.public.tickets_mv;

  catgroup | qtysold
 ----------+---------
  Concerts | 195444
    Shows  | 149905
```

È possibile gestire tabelle comuni in tutti i tenant di un cluster producer. È inoltre possibile condividere sottoinsiemi di dati filtrati in base alle colonne della dimensione, ad esempio `tenant_id` (`account_id` o `namespace_id`), con i cluster consumer. Per fare ciò, è possibile definire una vista sulla tabella di base con un filtro su queste colonne ID, ad esempio `current_aws_account = tenant_id`. Sul lato consumer, quando si esegue una query sulla vista, vengono visualizzate solo le righe idonee per il proprio account. A tale scopo, è possibile utilizzare le funzioni di contesto Amazon Redshift `current_aws_account` e `current_namespace`.

La seguente query restituisce l'ID account in cui risiede il cluster Amazon Redshift corrente. È possibile eseguire questa query se si è connessi ad Amazon Redshift.

```
select current_user, current_aws_account;

current_user | current_aws_account
-------------+--------------------
dwuser       |    111111111111
(1row)
```

La seguente query restituisce lo spazio dei nomi del cluster Amazon Redshift corrente. È possibile eseguire questa query se si è connessi al database.

```
select current_user, current_namespace;

current_user | current_namespace
-------------+--------------------------------------
```

```
dwuser       | 86b5169f-01dc-4a6f-9fbb-e2e24359e9a8
(1 row)
```

Aggiornamento incrementale per le viste materializzate in un datashare

Amazon Redshift supporta l'aggiornamento incrementale per le viste materializzate in un datashare consumer quando le tabelle di base vengono condivise. L'aggiornamento incrementale è un'operazione in cui Amazon Redshift identifica le modifiche nella tabella o nelle tabelle di base avvenute dopo l'aggiornamento precedente e aggiorna solo i record corrispondenti nella vista materializzata. Per ulteriori informazioni su questo comportamento, consulta CREATE MATERIALIZED VIEW.

Gestione dell'accesso alle operazioni API di condivisione dati con policy IAM

Per controllare l'accesso alle operazioni API di condivisione dati, utilizzare le policy basate su azioni IAM. Per ulteriori informazioni sulla gestione e sulla creazione di policy IAM personalizzate, consultare Gestione delle policy IAM nella Guida per l'utente di IAM.

Per informazioni sulle autorizzazioni necessarie per utilizzare le operazioni API di condivisione dei dati, consulta Autorizzazioni necessarie per utilizzare le operazioni dell'API di condivisione dei dati nella Guida alla gestione di Amazon Redshift.

Per rendere più sicura la condivisione dei dati tra account, è possibile utilizzare una chiave condizionale `ConsumerIdentifier` per operazioni API `AuthorizeDataShare` e `DeauthorizeDataShare`. In questo modo, puoi controllare in modo esplicito chi Account AWS può effettuare chiamate alle due operazioni API.

Puoi negare l'autorizzazione o la deautorizzazione della condivisione dei dati per qualsiasi consumatore che non sia il tuo account. A tale scopo, specifica il Account AWS numero nella policy IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Deny",
            "Action": [
                "redshift:AuthorizeDataShare",
```

```
                "redshift:DeauthorizeDataShare"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "redshift:ConsumerIdentifier": "555555555555"
                }
            }
        }
    ]
}
```

Puoi consentire a un produttore con un di DataShareArn **`testshare2`** condividerlo esplicitamente con un consumatore con uno Account AWS di 111122223333 nella policy IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "redshift:AuthorizeDataShare",
                "redshift:DeauthorizeDataShare"
            ],
            "Resource": "arn:aws:redshift:us-
east-1:666666666666:datashare:af06285e-8a45-4ee9-b598-648c218c8ff1/testshare2",
            "Condition": {
                "StringEquals": {
                    "redshift:ConsumerIdentifier": "111122223333"
                }
            }
        }
    ]
}
```

Esecuzione di query sulle unità di condivisione dati

Accesso ai dati condivisi in Amazon Redshift

È possibile individuare i dati condivisi utilizzando interfacce SQL standard, driver JDBC o ODBC e l'API dati. È inoltre possibile eseguire query sui dati con prestazioni elevate da strumenti di business

intelligence (BI) e analisi familiari. È possibile eseguire query facendo riferimento agli oggetti di altri database Amazon Redshift locali e remoti dal cluster a cui si dispone delle autorizzazioni di accesso.

È possibile farlo semplicemente rimanendo connessi ai database locali nel cluster. Quindi creare database consumer dalle unità di condivisione dati per utilizzare i dati condivisi.

Dopo averlo fatto, è possibile eseguire query tra database unendo i set di dati. È possibile eseguire query sugli oggetti nei database consumer utilizzando la notazione in 3 parti (*consumer_database_name*.*schema_name*.*table_name*). È inoltre possibile eseguire query utilizzando collegamenti di schema esterni agli schemi del database consumer. È possibile interrogare sia i dati locali che i dati condivisi da altri cluster all'interno della stessa query. Tale query può fare riferimento agli oggetti del database connesso corrente e da altri database non connessi, inclusi i database consumer creati da unità di condivisione dati.

Accesso ai metadati per le unità di condivisioni dati in Amazon Redshift

Per aiutare gli amministratori del cluster a individuare le unità di condivisione dati, Amazon Redshift fornisce una serie di viste di metadati per elencare le unità. Queste viste elencano le unità di condivisione dati create nel cluster e anche quelle ricevute da altri cluster all'interno dello stesso account, da altri account o altre regioni AWS . In queste viste vengono visualizzate le informazioni riportate di seguito.

- Unità di condivisione dati che vengono condivise e ricevute dai cluster
- Contenuto degli oggetti di database nella unità di condivisione dati, inclusi i metadati di condivisione di base, gli oggetti e i consumer

Utilizzare SVV_DATASHARES per visualizzare un elenco di tutte le unità di condivisione dati create nel cluster (in uscita) e condivise da altri (in entrata). Per ulteriori informazioni, consulta SVV_DATASHARES.

Utilizza SVV_DATASHARE_CONSUMERS per visualizzare un elenco di consumer di dati. Per ulteriori informazioni, consulta SVV_DATASHARE_CONSUMERS.

Utilizzare SVV_DATASHARE_OBJECTS per visualizzare un elenco di tutte le unità di condivisione dati create nel cluster (in uscita) e condivise da altri (in entrata). Per ulteriori informazioni, consulta SVV_DATASHARE_OBJECTS.

Integrazione della condivisione dei dati Amazon Redshift con gli strumenti di business intelligence

Per integrare la condivisione dei dati con gli strumenti di business intelligence (BI), consigliamo di utilizzare i driver JDBC o ODBC di Amazon Redshift.

I driver JDBC e ODBC di Amazon Redshift supportano l'operazione API `GetCatalogs` nei driver, che restituisce l'elenco di tutti i database, inclusi quelli creati dalle unità di condivisione dati. I driver supportano anche operazioni a valle, come `GetSchemas`,`GetTables` e così via, che restituiscono i dati da tutti i database restituiti da `GetCatalogs`. I driver forniscono questo supporto anche quando il catalogo non è specificato esplicitamente nella chiamata. Per ulteriori informazioni sui driver JDBC o ODBC, consulta Driver JDBC e ODBC per Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Non è possibile connettersi ai database consumer creati direttamente dalle unità di condivisione dati. Connettersi ai database locali nel cluster. Se si dispone di un'interfaccia utente di commutazione di connessione nello strumento, l'elenco dei database deve includere solo i database cluster locali. L'elenco dovrebbe escludere i database consumer creati dalle unità di condivisione dati per fornire la migliore esperienza. È possibile utilizzare un'opzione nella vista SVV_REDSHIFT_DATABASES per filtrare i database.

Monitoraggio e verifica della condivisione dati in Amazon Redshift

Verificando la condivisione dei dati, i producer possono tenere traccia dell'evoluzione dell'unità di condivisione dati. Ad esempio, il controllo aiuta a tenere traccia della creazione di condivisioni di dati, dell'aggiunta o della rimozione degli oggetti e della concessione o revoca delle autorizzazioni a cluster, account o regioni di Amazon Redshift. AWS AWS

Oltre alla verifica, producer e consumer tengono traccia dell'utilizzo delle unità di condivisione dati a vari livello di dettaglio, come account, cluster e oggetti. Per ulteriori informazioni sul monitoraggio dell'utilizzo e delle viste di verifica, consultare SVL_DATASHARE_CHANGE_LOG e SVL_DATASHARE_USAGE_PRODUCER.

Puoi monitorare le unità di condivisione dati eseguendo query sulle viste di sistema.

1. L'amministratore del cluster producer che desidera condividere i dati crea una unità di condivisione dati Amazon Redshift. L'amministratore del cluster producer aggiunge quindi gli oggetti del database necessari. Questi possono essere schemi, tabelle e viste per l'unità di condivisione dati e specifica un elenco di consumatori con cui condividere gli oggetti.

Utilizza le seguenti viste di sistema per visualizzare le viste consolidate per tenere traccia delle modifiche e dell'utilizzo delle unità di condivisione dati nei cluster producer e/o consumer:

- SYS_DATASHARE_CHANGE_LOG
- SYS_DATASHARE_USAGE_CONSUMER
- SYS_DATASHARE_USAGE_PRODUCER

Utilizzare le viste di sistema riportate di seguito per visualizzare gli oggetti dell'unità di condivisione dati e le informazioni sui consumer dei dati per le unità di condivisione dati in uscita:

- SVV_DATASHARES
- SVV_DATASHARE_CONSUMERS
- SVV_DATASHARE_OBJECTS

2. Gli amministratori del cluster consumer esaminano le unità di condivisione dati per le quali viene concesso l'utilizzo e riesaminano il contenuto di ogni unità visualizzando le unità di condivisione dati in ingresso tramite SVV_DATASHARES.

   Per utilizzare i dati condivisi, ogni amministratore del cluster consumer crea un database Amazon Redshift dall'unità di condivisione dati. L'amministratore assegna le autorizzazioni agli utenti e ai ruoli appropriati nel cluster consumer. Gli utenti e i ruoli possono elencare gli oggetti condivisi come parte delle query di metadati standard visualizzando le viste di sistema dei metadati seguenti e avviare immediatamente la query dei dati.

   - SVV_REDSHIFT_COLUMNS
   - SVV_REDSHIFT_DATABASE
   - SVV_REDSHIFT_FUNCTIONS
   - SVV_REDSHIFT_SCHEMAS
   - SVV_REDSHIFT_TABLES

   Per visualizzare gli oggetti degli schemi locali e condivisi di Amazon Redshift e degli schemi esterni, utilizzare le seguenti viste del sistema di metadati per eseguire le query.

   - SVV_ALL_COLUMNS
   - SVV_ALL_SCHEMAS
   - SVV_ALL_TABLES

Integrazione della condivisione dei dati di Amazon Redshift con AWS CloudTrail

La condivisione dei dati è integrata con. AWS CloudTrail CloudTrail è un servizio che fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon Redshift. CloudTrail acquisisce tutte le chiamate API per la condivisione dei dati come eventi. Le chiamate acquisite includono chiamate dalla AWS CloudTrail console e chiamate in codice alle operazioni di condivisione dei dati. Per ulteriori informazioni sull'integrazione di Amazon Redshift con AWS CloudTrail, consulta Logging with. CloudTrail

Per ulteriori informazioni su CloudTrail, consulta How CloudTrail works.

Gestione delle attività di condivisione dati

È possibile iniziare a condividere i dati utilizzando l'interfaccia SQL o la console Amazon Redshift.

Argomenti

- Gestione della condivisione dati tramite l'interfaccia SQL
- Gestione della condivisione dei dati mediante la console
- Gestione della condivisione dati con AWS CloudFormation
- Gestione della condivisione dei dati con operazioni di scrittura tramite la console (anteprima)

Gestione della condivisione dati tramite l'interfaccia SQL

Puoi condividere i dati a scopo di lettura tra diversi cluster Amazon Redshift all'interno o tra Account AWS o tra Regioni AWS.

Argomenti

- Condivisione dell'accesso in lettura ai dati all'interno di un Account AWS
- Condivisione dell'accesso in scrittura ai dati (anteprima)
- Condivisione dei dati tra Account AWS
- Condivisione dei dati tra Regioni AWS
- Condivisione di dati con licenza Amazon Redshift su AWS Data Exchange
- Lavorare con -managed datashares AWS Lake Formation

Condivisione dell'accesso in lettura ai dati all'interno di un Account AWS

È possibile condividere i dati a scopo di lettura tra diversi cluster Amazon Redshift all'interno di un account Account AWS.

Come condividere i dati a scopo di lettura come amministratore del cluster producer o proprietario del database

1. Creare le unità di condivisione dati nel cluster. Per ulteriori informazioni, consulta CREARE DATASHARE.

   ```
   CREATE DATASHARE salesshare;
   ```

   Le unità di condivisione dati possono essere create dall'utente con privilegi avanzati e dai proprietari di database del cluster. Ogni unità di condivisione dati è associata a un database durante la creazione. Solo gli oggetti di quel database possono essere condivisi in quella unità di condivisione dati. Sullo stesso database possono essere create più unità di condivisione dati con la stessa granularità di oggetti o con una granularità differente. Non vi è alcun limite sul numero di unità di condivisione dati che un cluster può creare.

   Per creare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Creazione di unità di condivisione dati.

2. Delegare le autorizzazioni per operare sull'unità di condivisione dati. Per ulteriori informazioni, consultare GRANT o REVOKE.

   L'esempio seguente concede le autorizzazioni a `dbuser` su `salesshare`.

   ```
   GRANT ALTER, SHARE ON DATASHARE salesshare TO dbuser;
   ```

   Gli utenti con privilegi avanzati del cluster e i proprietari dell'unità di condivisione dati possono concedere o revocare le autorizzazioni di modifica sull'unità di condivisione dati ad altri utenti.

3. Aggiungere oggetti o rimuovere oggetti dalle unità di condivisione dati. Per aggiungere oggetti a una unità di condivisione dati, aggiungere lo schema prima di aggiungere gli oggetti. Quando si aggiunge uno schema, Amazon Redshift non aggiunge tutti gli oggetti. Assicurarsi di aggiungerli esplicitamente. Per ulteriori informazioni, consulta ALTER DATASHARE.

   ```
   ALTER DATASHARE salesshare ADD SCHEMA PUBLIC;
   ALTER DATASHARE salesshare ADD TABLE public.tickit_sales_redshift;
   ```

```
ALTER DATASHARE salesshare ADD ALL TABLES IN SCHEMA PUBLIC;
```

A una unità di condivisione dati è possibile aggiungere anche le viste.

```
CREATE VIEW public.sales_data_summary_view AS SELECT * FROM
 public.tickit_sales_redshift;
ALTER DATASHARE salesshare ADD TABLE public.sales_data_summary_view;
```

Utilizzare ALTER DATASHARE per condividere schemi e tabelle, viste e funzioni in un determinato schema. Gli utenti con privilegi avanzati, i proprietari di unità di condivisione dati o gli utenti che hanno l'autorizzazione ALTER o ALL sull'unità di condivisione dati possono modificare l'unità in modo da aggiungere o rimuovere oggetti. Gli utenti devono disporre delle autorizzazioni per aggiungere o rimuovere oggetti dall'unità di condivisione dati. Gli utenti devono inoltre essere i proprietari degli oggetti o disporre delle autorizzazioni SELECT, USAGE o ALL sugli oggetti.

È inoltre possibile utilizzare GRANT per aggiungere oggetti al datashare. Questo esempio mostra come:

```
GRANT SELECT ON TABLE public.tickit_sales_redshift TO DATASHARE salesshare;
```

Questa sintassi è funzionalmente equivalente a. `ALTER DATASHARE salesshare ADD TABLE public.tickit_sales_redshift;`

Utilizzare la clausola INCLUDENEW per aggiungere nuove tabelle, viste o funzioni definite dall'utente (FDU) SQL future create in uno scherma specificato nell'unità di condivisione dati. Solo gli utenti con privilegi avanzati possono modificare questa proprietà per ogni coppia unità di condivisione dati-schema.

```
ALTER DATASHARE salesshare ADD SCHEMA PUBLIC;
ALTER DATASHARE salesshare SET INCLUDENEW = TRUE FOR SCHEMA PUBLIC;
```

Per aggiungere o rimuovere oggetti dalle unità di condivisione dati è possibile utilizzare la console Amazon Redshift. Per ulteriori informazioni, consultare Aggiunta di oggetti di unità di condivisione dati alle unità di condivisione dati, Rimozione di oggetti di unità di condivisione dati dalle unità di condivisione dati e Modifica delle unità di condivisione dati create nell'account.

4. Aggiungere o rimuovere i consumer dalle unità di condivisione dati. Nell'esempio seguente viene aggiunto lo spazio dei nomi del cluster consumer a `salesshare`. Lo spazio dei nomi è il

GUID dello spazio dei nomi del cluster consumer nell'account. Per ulteriori informazioni, consulta GRANT o REVOKE.

```
GRANT USAGE ON DATASHARE salesshare TO NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

È possibile concedere le autorizzazioni a un consumer dell'unità di condivisione dati solo in un'istruzione GRANT.

Gli utenti con privilegi avanzati per i cluster e i proprietari degli oggetti dell'unità di condivisione dati o gli utenti che dispongono dell'autorizzazione SHARE per l'unità di condivisione dati possono aggiungere o rimuovere i consumer da tale unità. Per fare ciò, usano GRANT USAGE o REVOKE USAGE

Per trovare lo spazio dei nomi del cluster attualmente visualizzato, è possibile utilizzare il comando SELECT CURRENT_NAMESPACE. Per trovare lo spazio dei nomi di un cluster diverso all'interno dello stesso Account AWS, vai alla pagina dei dettagli del cluster della console Amazon Redshift. In quella pagina, individuare il campo dello spazio dei nomi appena aggiunto.

È inoltre possibile utilizzare la console Amazon Redshift per aggiungere o rimuovere i consumer di dati dalle unità di condivisione dati. Per ulteriori informazioni, consultare Aggiunta di consumer di dati alle unità di condivisione dati e Rimozione dei consumer di dati dalle unità di condivisione dati.

5. (Facoltativo) Aggiungere le limitazioni di sicurezza all'unità di condivisione dati. L'esempio seguente mostra che il cluster consumer con un accesso IP pubblico è autorizzato a leggere l'unità di condivisione dati. Per ulteriori informazioni, consulta ALTER DATASHARE.

```
ALTER DATASHARE salesshare SET PUBLICACCESSIBLE = TRUE;
```

È possibile modificare le proprietà relative al tipo di consumer dopo la creazione dell'unità di condivisione dati. Ad esempio, è possibile definire che i cluster che desiderano utilizzare dati da una determinata unità di condivisione dati non possano essere accessibili pubblicamente. Le query provenienti da cluster di consumer che non soddisfano le limitazioni di sicurezza specificate nell'unità di condivisione dati vengono rifiutate al momento dell'esecuzione della query.

Per modificare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Modifica delle unità di condivisione dati create nell'account.

6. Elencare le unità di condivisione dati create nel cluster ed esaminare il contenuto dell'unità.

   Nell'esempio seguente sono riportate le informazioni di una unità di condivisione dati denominata `salesshare`. Per ulteriori informazioni, consulta DESC DATASHARE e SHOW DATASHARES.

```
DESC DATASHARE salesshare;


 producer_account  |          producer_namespace          | share_type | share_name
 | object_type |           object_name          |   include_new
------------------+--------------------------------------+------------
+------------+-------------+--------------------------------+-------------------
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_users_redshift   |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_venue_redshift   |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_category_redshift|
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_date_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_event_redshift   |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_listing_redshift |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_sales_redshift   |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | schema      | public                         |  t
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | view        | public.sales_data_summary_view |
```

   L'esempio seguente mostra le unità di condivisione dati in uscita in un cluster producer.

```
SHOW DATASHARES LIKE 'sales%';
```

   L'output è simile al seguente.

```
share_name | share_owner  | source_database | consumer_database | share_type |
 createdate      | is_publicaccessible  | share_acl | producer_account |
 producer_namespace
-----------+--------------+-----------------+-------------------+------------
+--------------------+---------------------+-----------+-----------------
+---------------------------------------
salesshare |    100       | dev             |                   |  OUTBOUND
 | 2020-12-09 02:27:08 |          True        |           |   123456789012   |
 13b8833d-17c6-4f16-8fe4-1a018f5ed00d
```

Per ulteriori informazioni, consulta DESC DATASHARE e SHOW DATASHARES.

È possibile utilizzare anche SVV_DATASHARES, SVV_DATASHARE_CONSUMERS e SVV_DATASHARE_OBJECTS per visualizzare le unità di condivisione dati, gli oggetti all'interno dell'unità di condivisione dati e i consumer dell'unità di condivisione dati.

7. Eliminare le unità di condivisione dati. Per ulteriori informazioni, consulta DROP DATASHARE.

   Gli oggetti dell'unità di condivisione dati possono essere eliminati in qualsiasi momento usando DROP DATASHARE. Le unità di condivisione dati possono essere eliminate dagli utenti con privilegi avanzati del cluster e dai proprietari dell'unità di condivisione dati.

   Nell'esempio seguente viene rimossa un'unità di condivisione dati denominata `salesshare`.

   ```
   DROP DATASHARE salesshare;
   ```

   Per eliminare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Eliminazione delle unità di condivisione dati create nell'account.

8. Utilizzare ALTER DATASHARE per rimuovere gli oggetti dalle unità di condivisione dati in qualsiasi punto dall'unità. Utilizzare REVOKE USAGE ON per revocare le autorizzazioni sull'unità di condivisione dati a su consumer. Revoca le autorizzazioni USAGE sugli oggetti all'interno di una unità di condivisione dati e disabilita immediatamente l'accesso a tutti i cluster di consumer. L'elenco delle unità di condivisione dati e delle query dei metadati, ad esempio l'elenco dei database e delle tabelle, non restituisce gli oggetti condivisi dopo la revoca dell'accesso.

   ```
   ALTER DATASHARE salesshare REMOVE TABLE public.tickit_sales_redshift;
   ```

Per modificare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Modifica delle unità di condivisione dati create nell'account.

9. Revocare l'accesso all'unità di condivisione dati dagli spazi dei nomi se non si desidera più condividere i dati con i consumer.

```
REVOKE USAGE ON DATASHARE salesshare FROM NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

Per modificare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Modifica delle unità di condivisione dati create nell'account.

Come condividere i dati a scopo di lettura come amministratore del cluster consumer

1. Elencare le unità di condivisione dati rese disponibili e visualizzare il contenuto delle unità di condivisione dati. Per ulteriori informazioni, consulta DESC DATASHARE e SHOW DATASHARES.

   Nell'esempio seguente vengono visualizzate le informazioni relative alle unità di condivisione dati in ingresso di uno spazio dei nomi producer specificato. Quando si esegue DESC DATASHARE come amministratore del cluster consumer, è necessario specificare l'opzione NAMESPACE per visualizzare le unità di condivisione dati in ingresso.

```
DESC DATASHARE salesshare OF NAMESPACE '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';


 producer_account  |          producer_namespace          | share_type | share_name
 | object_type |           object_name           |   include_new
------------------+--------------------------------------+------------
+------------+-------------+----------------------------------+------------------
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_users_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_venue_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_category_redshift |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_date_redshift     |
```

```
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_event_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_listing_redshift  |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_sales_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | schema      | public                          |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | view        | public.sales_data_summary_view  |
```

Solo gli utenti con privilegi avanzati per il cluster possono completare questa opzione. È possibile utilizzare anche SVV_DATASHARES per visualizzare le unità di condivisione dati e SVV_DATASHARE_OBJECTS per visualizzare gli oggetti all'interno dell'unità di condivisione dati.

L'esempio seguente mostra le unità di condivisione dati in uscita in un cluster producer.

```
SHOW DATASHARES LIKE 'sales%';


 share_name | share_owner | source_database | consumer_database | share_type
 | createdate | is_publicaccessible | share_acl | producer_account |
 producer_namespace
------------+-------------+-----------------+-------------------+------------
+------------+---------------------+-----------+------------------
+--------------------------------------
 salesshare |             |                 |                   | INBOUND
    |            |         t           |           |   123456789012   |
 13b8833d-17c6-4f16-8fe4-1a018f5ed00d
```

2. In qualità di utente di database con privilegi avanzati, puoi creare database locali che fanno riferimento alle unità di condivisione dati. Per ulteriori informazioni, consulta CREATE DATABASE.

```
CREATE DATABASE sales_db FROM DATASHARE salesshare OF NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

Se desideri un controllo più granulare sull'accesso agli oggetti nel database locale, utilizza la clausola WITH PERMISSIONS durante la creazione del database. In tal modo puoi assegnare le autorizzazioni per gli oggetti del database nel passaggio 4.

```
CREATE DATABASE sales_db WITH PERMISSIONS FROM DATASHARE salesshare OF NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

È possibile vedere i database creati dall'unità di condivisione dati eseguendo una query sulla vista SVV_REDSHIFT_DATABASE. Non è possibile connettersi a questi database creati da unità di condivisione dati e sono di sola lettura. Tuttavia, è possibile connettersi a un database locale nel cluster consumer ed eseguire una query tra database per eseguire query sui dati dei database creati dalle unità di condivisione dati. Non è possibile creare una unità di condivisione dati sugli oggetti di database creati da una unità di condivisione dati esistente. Tuttavia, è possibile copiare i dati in una tabella separata nel cluster consumer, eseguire qualsiasi elaborazione necessaria e quindi condividere i nuovi oggetti creati.

Per creare i database dalle unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Creazione di database da unità di condivisione dati.

3. (Facoltativo) Creare schemi esterni per fare riferimento e assegnare autorizzazioni granulari a schemi specifici nel database consumer importato nel cluster consumer. Per ulteriori informazioni, consulta CREATE EXTERNAL SCHEMA.

```
CREATE EXTERNAL SCHEMA sales_schema FROM REDSHIFT DATABASE 'sales_db' SCHEMA
 'public';
```

4. Concedi ai ruoli e agli utenti nel cluster consumer le autorizzazioni per i database e i riferimenti allo schema creati dalle unità di condivisione dati in base alle esigenze. Per ulteriori informazioni, consulta GRANT o REVOKE.

```
GRANT USAGE ON DATABASE sales_db TO Bob;
```

```
GRANT USAGE ON SCHEMA sales_schema TO ROLE Analyst_role;
```

Se hai creato il database senza la clausola WITH PERMISSIONS, puoi assegnare agli utenti e ai ruoli solo le autorizzazioni per l'intero database creato dall'unità di condivisione dati. In alcuni casi, sono necessari controlli a grana fine su un sottoinsieme di oggetti di database creati dall'unità di condivisione dati. In tal caso, è possibile creare un riferimento allo schema esterno che punti a schemi specifici nell'unità di condivisione dati (come descritto nel passaggio precedente) e fornire autorizzazioni granulari a livello di schema.

È inoltre possibile creare viste ad associazione tardiva sugli oggetti condivisi e utilizzarle per assegnare autorizzazioni granulari. È inoltre possibile considerare che i cluster producer creino ulteriori unità di condivisione dati con la granularità richiesta.

Se hai creato il database con la clausola WITH PERMISSIONS nel passaggio 2, devi assegnare le autorizzazioni per gli oggetti nel database condiviso. Un utente con solo l'autorizzazione USAGE non può accedere a nessun oggetto in un database creato con la clausola WITH PERMISSIONS finché non ottiene le autorizzazioni aggiuntive a livello di oggetto.

```
GRANT SELECT ON sales_db.public.tickit_sales_redshift to Bob;
```

5. Eseguire una query sui dati negli oggetti condivisi nelle unità di condivisione dati.

   Gli utenti e i ruoli con autorizzazioni per i database consumer e gli schemi nei cluster consumer possono esplorare e navigare tra i metadati di qualsiasi oggetto condiviso. Possono inoltre esplorare e navigare tra oggetti locali in un cluster consumer. Per fare ciò, utilizzano i driver JDBC o ODBC o le viste SVV_ALL e SVV_REDSHIFT.

   I cluster producer potrebbero avere molti schemi nel database, nelle tabelle e nelle viste all'interno di ogni schema. Gli utenti sul lato consumer possono vedere solo il sottoinsieme di oggetti che sono resi disponibili attraverso l'unità di condivisione dati. Questi utenti non possono visualizzare gli interi metadati dal cluster producer. Questo approccio consente di fornire un controllo granulare della sicurezza dei metadati con la condivisione dei dati.

   È possibile continuare a connettersi ai database locali nel cluster. Ma ora, è anche possibile leggere dai database e dagli schemi creati dall'unità di condivisione dati utilizzando la notazione database.schema.tabella in tre parti. È possibile eseguire query che si estendono su tutti i database visibili. Questi possono essere database locali sul cluster o database creati dalle unità di condivisione dati. I cluster consumer non possono connettersi ai database creati dalle unità di condivisione dati.

   È possibile accedere ai dati mediante la qualifica completa. Per ulteriori informazioni, consulta Esempi di utilizzo di una query tra database.

```
SELECT * FROM sales_db.public.tickit_sales_redshift ORDER BY 1,2 LIMIT 5;

 salesid | listid | sellerid | buyerid | eventid | dateid | qtysold | pricepaid |
 commission |      saletime
```

```
---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------
       1 |      1 |    36861 |   21191 |    7872 |   1875 |       4 |    728.00 |
   109.20 | 2008-02-18 02:36:48
       2 |      4 |     8117 |   11498 |    4337 |   1983 |       2 |     76.00 |
    11.40 | 2008-06-06 05:00:16
       3 |      5 |     1616 |   17433 |    8647 |   1983 |       2 |    350.00 |
    52.50 | 2008-06-06 08:26:17
       4 |      5 |     1616 |   19715 |    8647 |   1986 |       1 |    175.00 |
    26.25 | 2008-06-09 08:38:52
       5 |      6 |    47402 |   14115 |    8240 |   2069 |       2 |    154.00 |
    23.10 | 2008-08-31 09:17:02
```

È possibile utilizzare le istruzioni SELECT solo su oggetti condivisi. Tuttavia, è possibile creare tabelle nel cluster consumer eseguendo una query sui dati degli oggetti condivisi in un database locale diverso.

Oltre alle query, i consumer possono creare viste su oggetti condivisi. Sono supportate solo le viste ad associazione tardiva o le viste materializzate. Amazon Redshift non supporta le viste regolari sui dati condivisi. Le viste create dai consumer possono estendersi su più database locali o database creati dalle unità di condivisione dati. Per ulteriori informazioni, consulta CREATE VIEW.

```
// Connect to a local cluster database

// Create a view on shared objects and access it.
CREATE VIEW sales_data
AS SELECT *
FROM sales_db.public.tickit_sales_redshift
WITH NO SCHEMA BINDING;

SELECT * FROM sales_data;
```

Condivisione dell'accesso in scrittura ai dati (anteprima)

Puoi condividere oggetti di database per le letture e le scritture tra diversi cluster Amazon Redshift o gruppi di lavoro Serverless Amazon Redshift all'interno dello Account AWS stesso, tra account e regioni. Le procedure in questo argomento mostrano come configurare la condivisione dei dati che include le autorizzazioni di scrittura. È possibile concedere autorizzazioni quali SELECT, INSERT e UPDATE per tabelle diverse e USAGE e CREATE per schemi. I dati sono attivi e disponibili per tutti

i warehouse non appena viene eseguita una transazione di scrittura. Gli amministratori degli account Producer possono determinare se i namespace o le aree geografiche specifici devono essere resi di sola lettura o se consentire l'accesso ai dati. read-and-write

Le sezioni che seguono mostrano come configurare la condivisione dei dati. Le procedure presuppongono che si utilizzi un database in un cluster con provisioning o in un gruppo di lavoro Amazon Redshift serverless.

Condivisione dei dati in sola lettura e condivisione dei dati in lettura e scrittura

In precedenza, gli oggetti nelle unità di condivisione dati venivano solo letti in tutte le circostanze. La scrittura in un oggetto di un'unità di condivisione dati è una nuova funzionalità. Gli oggetti nelle unità di condivisione dati sono abilitati alla scrittura solo quando un producer assegna specificamente i privilegi di scrittura come INSERT o CREATE agli oggetti dell'unità di condivisione dati. Inoltre, per la condivisione tra più account, un produttore deve autorizzare il datashare per le scritture e il consumatore deve associare cluster e gruppi di lavoro specifici per le scritture. I dettagli sono riportati nelle sezioni successive di questo argomento.

Autorizzazioni che puoi concedere alle condivisioni di dati (anteprima)

I diversi tipi di oggetti e le varie autorizzazioni che puoi assegnare in un contesto di condivisione dei dati.

Schemi:

- USAGE
- CREATE

Tabelle:

- SELECT
- INSERT
- UPDATE
- DELETE
- TRUNCATE
- DROP
- REFERENCES

Funzioni:

- EXECUTE

Database:

- CREATE

Requisiti e limitazioni per l'unità di condivisione dati in anteprima

- Connessioni: è necessario connettersi direttamente a un database di datashare o eseguire il comando USE per scrivere su datashare. Tuttavia, presto abiliteremo la possibilità di eseguire questa operazione usando una notazione in tre parti.
- Disponibilità: è necessario utilizzare gruppi di lavoro Serverless, cluster ra3.4xl o cluster ra3.16xl per utilizzare questa funzionalità. È previsto il supporto per i cluster ra3.xlplus.
- Metadata Discovery: se sei un consumatore connesso direttamente a un database di datashare tramite i driver JDBC, ODBC o Python Redshift, puoi visualizzare i dati del catalogo nei seguenti modi:
  - Comandi SQL SHOW.
  - Query su tabelle e viste information_schema.
  - Query su viste SVV dei metadati.
- Data API: non è possibile connettersi ai database di datashare tramite l'API Data. Il supporto per questa operazione sarà presto disponibile.
- Visibilità delle autorizzazioni: i consumatori non possono vedere le autorizzazioni concesse alle condivisioni di dati. Aggiungeremo presto questa operazione.
- Crittografia: per la condivisione dei dati tra account, è necessario crittografare sia il cluster di produttori che quello di consumatori.
- Livello di isolamento: il livello di isolamento del database deve essere l'isolamento delle istantanee per consentire ad altri gruppi di lavoro e cluster Serverless di scrivervi.
- Operazioni automatiche: i consumatori che scrivono su oggetti datashare non attiveranno un'operazione di analisi automatica. Di conseguenza, il producer deve eseguire manualmente l'analisi dopo l'inserimento dei dati nella tabella per aggiornare le statistiche della tabella. In caso contrario, i piani di query potrebbero non essere ottimali.

- Interrogazioni e transazioni con più dichiarazioni: le query con più istruzioni al di fuori di un blocco di transazioni non sono attualmente supportate. Di conseguenza, se utilizzi un editor di query come dbeaver e hai più query di scrittura, devi racchiudere le query in un'istruzione di transazione BEGIN... END esplicita.

Istruzioni SQL supportate

Queste istruzioni sono supportate per la versione di anteprima pubblica della condivisione di dati con le operazioni di scrittura:

- BEGIN | START TRANSACTION
- END | COMMIT | ROLLBACK
- COPY senza COMPUPDATE
- { CREATE | DROP } SCHEMA
- { CREATE | DROP | SHOW } TABLE
- CREATE TABLE table_name AS
- DELETE
- { GRANT | REVOKE } privilege_name ON OBJECT_TYPE object_name TO consumer_user
- INSERT
- SELECT
- INSERT INTO SELECT
- TRUNCATE
- UPDATE
- Colonne con tipo di dati Super

Tipi di istruzioni non supportati: i seguenti tipi non sono supportati:

- Query con più istruzioni sui warehouse consumer che scrivono nei producer.
- Query di dimensionamento simultaneo con scrittura dai consumer ai producer.
- Processi di copia automatica con scrittura dai consumer ai producer.
- Processi di streaming con scrittura dai consumer ai producer.
- Consumer che creano tabelle di integrazione Zero-ETL su cluster producer. Per ulteriori informazioni sulle integrazioni Zero-ETL, consulta Utilizzo delle integrazioni Zero-ETL.

- Scrittura in una tabella con una chiave di ordinamento interleaved.

Condivisione di dati all'interno di un account con autorizzazioni di scrittura come amministratore dell'account produttore (anteprima)

In precedenza, gli oggetti nelle unità di condivisione dati venivano solo letti in tutte le circostanze. La scrittura in un oggetto di un'unità di condivisione dati è una nuova funzionalità. Gli oggetti nelle unità di condivisione dati sono abilitati alla scrittura solo quando un producer assegna specificamente i privilegi di scrittura come INSERT o CREATE agli oggetti dell'unità di condivisione dati. I dettagli sono riportati nelle sezioni successive di questo argomento.

Per la documentazione esistente delle unità di condivisione dati di sola lettura, consulta Condivisione di dati tra cluster in Amazon Redshift.

Per iniziare la condivisione dei dati, l'amministratore del producer crea un'unità di condivisione dati e aggiunge gli oggetti:

1. Il proprietario o l'utente con privilegi avanzati del database producer crea un'unità di condivisione dati. L'unità di condivisione dati è un container logico di oggetti, autorizzazioni e consumer del database. I consumer sono cluster o spazi dei nomi Amazon Redshift serverless presenti nel tuo account e in altri. Ogni unità di condivisione dati è associata al database in cui è stata creata ed è possibile aggiungere solo gli oggetti di quel database. Il seguente comando crea un'unità di condivisione dati:

   ```
   CREATE DATASHARE my_datashare [PUBLICACCESSIBLE = TRUE];
   ```

   L'impostazione PUBLICACCESSIBLE = TRUE consente ai consumer di eseguire query sull'unità di condivisione dati da cluster accessibili pubblicamente e gruppi di lavoro con provisioning. Disattivala o impostala esplicitamente su false se non vuoi consentirla.

   Il proprietario dell'unità di condivisione dati deve assegnare l'autorizzazione USAGE per gli schemi che desidera aggiungere all'unità di condivisione dati. Il comando GRANT è nuovo. Viene utilizzato per assegnare varie azioni per lo schema, tra cui CREATE e USAGE. Gli schemi contengono oggetti condivisi:

   ```
   CREATE SCHEMA myshared_schema1;
   CREATE SCHEMA myshared_schema2;

   GRANT USAGE ON SCHEMA myshared_schema1 TO DATASHARE my_datashare;
   ```

```
GRANT CREATE, USAGE ON SCHEMA myshared_schema2 TO DATASHARE my_datashare;
```

In alternativa, l'amministratore può continuare a eseguire i comandi ALTER per aggiungere uno schema all'unità di condivisione dati. Quando uno schema viene aggiunto in questo modo, vengono assegnate solo le autorizzazioni USAGE.

```
ALTER DATASHARE my_datashare ADD SCHEMA myshared_schema1;
```

2. Dopo aver aggiunto gli schemi, l'amministratore può assegnare le autorizzazioni dell'unità di condivisione dati per gli oggetti dello schema, che possono essere autorizzazioni in lettura e scrittura. L'esempio GRANT ALL mostra come assegnare tutte le autorizzazioni.

```
GRANT SELECT, INSERT ON TABLE myshared_schema1.table1, myshared_schema1.table2,
 myshared_schema2.table1
TO DATASHARE my_datashare;

GRANT ALL ON TABLE myshared_schema1.table4 TO DATASHARE my_datashare;
```

È possibile continuare a eseguire comandi come ALTER DATASHARE per aggiungere le tabelle. In tal caso, vengono assegnate solo le autorizzazioni SELECT per gli oggetti aggiunti.

```
ALTER DATASHARE my_datashare ADD TABLE myshared_schema1.table1,
 myshared_schema1.table2, myshared_schema2.table1;
```

3. L'amministratore assegna l'utilizzo dell'unità di condivisione dati a uno spazio dei nomi specifico dell'account. Puoi trovare l'ID dello spazio dei nomi come parte dell'ARN nella pagina dei dettagli del cluster, nella pagina dei dettagli dello spazio dei nomi Amazon Redshift serverless o eseguendo il comando `SELECT current_namespace;`. Per ulteriori informazioni, consulta CURRENT_NAMESPACE.

```
GRANT USAGE ON DATASHARE my_datashare TO NAMESPACE '86b5169f-012a-234b-9fbb-
e2e24359e9a8';
```

Condivisione delle autorizzazioni di scrittura sui dati tra account (anteprima)

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a

modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Se non hai ancora creato un datashare sulla traccia PREVIEW_2023, vai a Condivisione dell'accesso alla scrittura ai dati (anteprima) per iniziare.

Associazione di dati condivisi come amministratore della sicurezza dei dati dei consumer (anteprima)

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Se non hai ancora creato un datashare sulla traccia PREVIEW_2023, vai a Condivisione, scrittura, accesso ai dati (anteprima) per iniziare.

Prerequisiti: i passaggi descritti in questa sezione vengono eseguiti dopo che l'amministratore del producer assegna azioni specifiche per gli oggetti del database condiviso e, se l'unità di condivisione dati è condivisa con un altro account, l'amministratore della sicurezza del producer ne autorizza l'accesso.

L'amministratore della sicurezza dei consumer determina quanto segue:

- Indica se tutti gli spazi dei nomi di un account, gli spazi dei nomi in regioni specifiche dell'account o spazi dei nomi specifici hanno accesso all'unità di condivisione dati.
- Indica se gli spazi dei nomi hanno accesso all'unità di condivisione dati, indipendentemente dal fatto che abbiano o meno le autorizzazioni di scrittura.

L'amministratore della sicurezza dei consumer può associare l'unità di condivisione dati usando la console, la CLI o tramite API. Con la CLI, l'amministratore utilizza il seguente comando:

```
associate-data-share-consumer
--data-share-arn <value>
```

```
--consumer-identifier <value>
[--allow-writes | --no-allow-writes]
```

Per ulteriori informazioni sul comando, consulta associate-data-share-consumer.

L'amministratore della sicurezza dei consumer deve impostare esplicitamente `allow-writes` su true quando associa un'unità di condivisione dati a uno spazio dei nomi, per consentire l'uso dei comandi INSERT e UPDATE. In caso contrario, gli utenti possono eseguire solo operazioni di lettura, come i privilegi SELECT, USAGE o EXECUTE.

È possibile modificare l'associazione di uno spazio dei nomi per un'unità di condivisione dati chiamando nuovamente `associate-data-share-consumer` con un valore diverso. La precedente associazione viene sovrascritta dalla nuova associazione, quindi se originariamente associ e imposti `allow-writes`, quindi associ e specifichi `no-allow-writes`, o semplicemente non specifichi un valore, al consumer verranno revocate le autorizzazioni di scrittura.

Autorizzazione delle unità di condivisione dati per le operazioni di scrittura come amministratore di sicurezza producer (anteprima)

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Se non hai ancora creato un datashare sulla traccia PREVIEW_2023, vai a Condivisione, scrittura, accesso ai dati (anteprima) per iniziare.

> **Note**
>
> Questo argomento si applica solo per l'unità di condivisione dati condivisa tra account.

L'amministratore della sicurezza dei producer determina quanto segue:

- Indica se un altro account può avere accesso o meno all'unità di condivisione dati.

- Indica se un account ha accesso all'unità di condivisione dati, indipendentemente dal fatto che disponga o meno delle autorizzazioni di scrittura.

Per autorizzare un'unità di condivisione dati sono necessarie le seguenti autorizzazioni IAM:

redshift: Condividi AuthorizeData

Puoi autorizzare l'utilizzo e le operazioni di scrittura utilizzando una chiamata della CLI o l'API:

```
authorize-data-share
--data-share-arn <value>
--consumer-identifier <value>
[--allow-writes | --no-allow-writes]
```

Per ulteriori informazioni sul comando, consulta authorize-data-share.

L'identificatore del consumer può essere:

- Un ID di AWS account a dodici cifre.
- L'ARN dell'identificatore dello spazio dei nomi.

Tieni presente che le autorizzazioni di scrittura non vengono assegnate nella fase di autorizzazione. L'autorizzazione di un'unità di condivisione dati per le operazioni di scrittura consente all'account di disporre solo delle autorizzazioni di scrittura assegnate dall'amministratore dell'unità di condivisione dati. Se un amministratore non consente le operazioni di scrittura, le uniche autorizzazioni disponibili per il consumer specifico sono SELECT, USAGE ed EXECUTE.

È possibile modificare l'autorizzazione di un consumer dell'unità di condivisione dati chiamando nuovamente `authorize-data-share`, ma con un valore diverso. La precedente autorizzazione viene sovrascritta dalla nuova autorizzazione. Pertanto, se originariamente autorizzi e consenti le operazioni di scrittura, quindi autorizzi nuovamente e specifichi `no-allow-writes` o semplicemente non specifichi un valore, al consumer verranno revocate le autorizzazioni di scrittura.

Regioni in cui è disponibile la condivisione dei dati (anteprima)

> Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di

produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Se non hai ancora creato un datashare sulla traccia PREVIEW_2023, vai a Condivisione, scrittura, accesso ai dati (anteprima) per iniziare.

Le seguenti regioni dispongono della condivisione dei dati in anteprima:

- Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Stati Uniti orientali (Ohio) (us-east-2)
- Stati Uniti occidentali (Oregon) (us-west-2)
- Asia Pacifico (Tokyo) (ap-northeast-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Stoccolma) (eu-north-1)

Condivisione dei dati tra Account AWS

Puoi condividere i dati a scopo di lettura tra Account AWS. La condivisione dei dati tra di loro Account AWS funziona in modo simile alla condivisione dei dati all'interno di un account. La differenza è che nella condivisione dei dati tra Account AWSè richiesto un handshake bidirezionale. Gli amministratori di un account producer possono autorizzare gli account consumer ad accedere alle unità di condivisione dati o scegliere di non autorizzare alcun accesso. Per utilizzare una condivisione dati autorizzata, un amministratore dell'account consumer può associare la condivisione dati. L'amministratore può associare il datashare a un intero Account AWS o a cluster specifici nell'account consumer oppure rifiutare il datashare. Per ulteriori informazioni sulla condivisione dei dati all'interno di un account, consultare Condivisione dell'accesso in lettura ai dati all'interno di un Account AWS.

Un'unità di condivisione dati può avere consumer di dati che sono spazi dei nomi del cluster nello stesso account o diversi Account AWS. Non è necessario creare unità di condivisione dati separate per la condivisione all'interno di un account e la condivisione tra account.

Per la condivisione dei dati tra account, sia il cluster producer che quello consumer devono essere crittografati.

Quando condividono i dati con Account AWS, gli amministratori del cluster di produttori li condividono come entità. Account AWS Un amministratore del cluster consumer può decidere quali spazi dei nomi del cluster nell'account consumer possono ottenere l'accesso a una unità di condivisione dati.

Argomenti

Operazioni dell'amministratore del cluster producer

Se si è un amministratore del cluster producer o un proprietario di database, completare la seguente procedura:

1. Creare unità di condivisione dati nel cluster e aggiungere oggetti di unità di condivisione dati alle unità. Per la procedura dettagliati di come creare unità di condivisione dati e come aggiungere oggetti di unità di condivisione dati alle unità, consultare Condivisione dell'accesso in lettura ai dati all'interno di un Account AWS. Per informazioni su CREATE DATASHARE e ALTER DATASHARE, consultare CREARE DATASHARE e ALTER DATASHARE.

   Nell'esempio seguente vengono aggiunti diversi oggetti di unità di condivisione dati all'unità di condivisione dati `salesshare`.

   ```
   -- Add schema to datashare
   ALTER DATASHARE salesshare ADD SCHEMA PUBLIC;

   -- Add table under schema to datashare
   ALTER DATASHARE salesshare ADD TABLE public.tickit_sales_redshift;

   -- Add view to datashare
   ALTER DATASHARE salesshare ADD TABLE public.sales_data_summary_view;

   -- Add all existing tables and views under schema to datashare (does not include
    future table)
   ALTER DATASHARE salesshare ADD ALL TABLES in schema public;
   ```

   Per creare o modificare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Creazione di unità di condivisione dati e Modifica delle unità di condivisione dati create nell'account.
2. Delegare le autorizzazioni per operare sull'unità di condivisione dati. Per ulteriori informazioni, consultare GRANT o REVOKE.

   L'esempio seguente concede le autorizzazioni a `dbuser` su `salesshare`.

```
GRANT ALTER, SHARE ON DATASHARE salesshare TO dbuser;
```

Gli utenti con privilegi avanzati del cluster e i proprietari dell'unità di condivisione dati possono concedere o revocare le autorizzazioni di modifica sull'unità di condivisione dati ad altri utenti.

3. Aggiungere o rimuovere i consumer dalle unità di condivisione dati. Nell'esempio seguente viene aggiunto l'ID Account AWS a `salesshare`. Per ulteriori informazioni, consulta GRANT o REVOKE.

   ```
   GRANT USAGE ON DATASHARE salesshare TO ACCOUNT '123456789012';
   ```

   È possibile concedere le autorizzazioni a un consumer di dati solo in un'istruzione GRANT.

   Gli utenti con privilegi avanzati dei cluster e i proprietari degli oggetti dell'unità di condivisione dati o gli utenti che dispongono dell'autorizzazione SHARE sull'unità di condivisione dati possono aggiungere o rimuovere i consumer da tale unità. Per fare ciò, usano GRANT USAGE o REVOKE USAGE

   È inoltre possibile utilizzare la console Amazon Redshift per aggiungere o rimuovere i consumer di dati dalle unità di condivisione dati. Per ulteriori informazioni, consulta Aggiunta di consumer di dati alle unità di condivisione dati e Rimozione dei consumer di dati dalle unità di condivisione dati.

4. (Facoltativo) Revoca l'accesso al datashare da Account AWS se non desideri più condividere i dati con i consumatori.

   ```
   REVOKE USAGE ON DATASHARE salesshare FROM ACCOUNT '123456789012';
   ```

Se si è un amministratore di account producer, completare la seguente procedura:

Dopo aver concesso l'utilizzo a, lo stato del datashare è Account AWS. `pending_authorization` L'amministratore dell'account producer deve autorizzare le unità di condivisione dati utilizzando la console Amazon Redshift e scegliere i consumer di dati.

Accedi a https://console.aws.amazon.com/redshiftv2/. Quindi scegliere quali consumer di dati autorizzare ad accedere alle unità di condivisione dati o da cui rimuovere l'autorizzazione. I consumer di dati autorizzati ricevono notifiche per intraprendere azioni sulle unità di condivisione dati. Se si aggiunge uno spazio dei nomi del cluster come consumer dati, non è necessario eseguire l'autorizzazione. Una volta autorizzati i consumer di dati, questi possono accedere agli oggetti

dell'unità di condivisione dati e creare un database consumer per eseguire le query sui dati. Per ulteriori informazioni, consulta Autorizzazione o rimozione dell'autorizzazione dalle unità di condivisione dati.

Operazioni dell'amministratore dell'account consumer

Se si è un amministratore di account consumer, completare la seguente procedura:

Per associare una o più condivisioni di dati condivise da altri account a namespace di cluster interi Account AWS o specifici nel tuo account, usa la console Amazon Redshift.

Accedi a https://console.aws.amazon.com/redshiftv2/. Quindi, associa uno o più datashare condivisi da altri account ai namespace del cluster interi Account AWS o specifici del tuo account. Per ulteriori informazioni, consulta Associazione delle unità di condivisione dati.

Dopo aver associato lo spazio dei nomi del cluster Account AWS o uno specifico, i datashare diventano disponibili per l'utilizzo. L'associazione dell'unità di condivisione dati può essere modificata in qualsiasi momento. Quando si modifica l'associazione da un singolo namespace di cluster a uno, Amazon Account AWS Redshift sovrascrive i namespace del cluster con le informazioni. Account AWS Quando si modifica l'associazione da uno spazio dei nomi di cluster Account AWS a uno specifico, Amazon Redshift sovrascrive le informazioni con le Account AWS informazioni sullo spazio dei nomi del cluster. Tutti gli spazi dei nomi del cluster nell'account ottengono l'accesso ai dati.

Operazioni dell'amministratore del cluster consumer

Se si è un amministratore di cluster consumer, completare la seguente procedura:

1. Elencare le unità di condivisione dati rese disponibili e visualizzare il contenuto delle unità di condivisione dati. Il contenuto delle unità di condivisione dati è disponibile solo quando l'amministratore del cluster producer ha autorizzato le unità di condivisione dati e l'amministratore del cluster consumer ha accettato e associato le unità. Per ulteriori informazioni, consulta DESC DATASHARE e SHOW DATASHARES.

   Nell'esempio seguente vengono visualizzate le informazioni relative alle unità di condivisione dati in ingresso di uno spazio dei nomi producer specificato. Quando si esegue DESC DATASHARE come amministratore del cluster consumer, è necessario specificare l'opzione NAMESPACE e ID account per visualizzare le unità di condivisione dati in ingresso. Per le condivisioni di dati in uscita, specificare il nome dell'unità di condivisione dati.

   ```
   SHOW DATASHARES LIKE 'sales%';
   ```

```
share_name | share_owner | source_database | consumer_database | share_type
 | createdate | is_publicaccessible | share_acl | producer_account |
 producer_namespace
-----------+-------------+-----------------+-------------------+-----------
+------------+---------------------+-----------+------------------
+--------------------------------------
salesshare |             |                 |                   | INBOUND    |
            |        t            |           | 123456789012    | 'dd8772e1-
d792-4fa4-996b-1870577efc0d'
```

```
DESC DATASHARE salesshare OF ACCOUNT '123456789012' NAMESPACE 'dd8772e1-
d792-4fa4-996b-1870577efc0d';


 producer_account |          producer_namespace          | share_type | share_name |
 object_type |           object_name
-----------------+--------------------------------------+------------+------------
+-------------+---------------------------------
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d | INBOUND    | salesshare |
 table       | public.tickit_users_redshift
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d | INBOUND    | salesshare |
 table       | public.tickit_venue_redshift
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d | INBOUND    | salesshare |
 table       | public.tickit_category_redshift
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d | INBOUND    | salesshare |
 table       | public.tickit_date_redshift
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d | INBOUND    | salesshare |
 table       | public.tickit_event_redshift
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d | INBOUND    | salesshare |
 table       | public.tickit_listing_redshift
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d | INBOUND    | salesshare |
 table       | public.tickit_sales_redshift
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d | INBOUND    | salesshare |
 schema      | public
(8 rows)
```

Solo gli utenti con privilegi avanzati per il cluster possono completare questa opzione. È possibile utilizzare anche SVV_DATASHARES per visualizzare le unità di condivisione dati e SVV_DATASHARE_OBJECTS per visualizzare gli oggetti all'interno dell'unità di condivisione dati.

L'esempio seguente mostra le unità di condivisione dati in uscita in un cluster producer.

```
SELECT * FROM SVV_DATASHARES WHERE share_name LIKE 'sales%';

share_name | share_owner | source_database | consumer_database | share_type
 | createdate | is_publicaccessible | share_acl | producer_account |
 producer_namespace
-----------+-------------+-----------------+-------------------+-----------
+------------+---------------------+-----------+-----------------
+--------------------------------------
salesshare |             |                 |                   | INBOUND    |
            |        t            |           | 123456789012      | 'dd8772e1-
d792-4fa4-996b-1870577efc0d'
```

```
SELECT * FROM SVV_DATASHARE_OBJECTS WHERE share_name LIKE 'sales%';
 share_type | share_name | object_type |           object_name           |
 producer_account |          producer_namespace
------------+------------+-------------+---------------------------------
+------------------+--------------------------------------
 INBOUND    | salesshare | table       | public.tickit_users_redshift    |
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d
 INBOUND    | salesshare | table       | public.tickit_venue_redshift    |
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d
 INBOUND    | salesshare | table       | public.tickit_category_redshift |
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d
 INBOUND    | salesshare | table       | public.tickit_date_redshift     |
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d
 INBOUND    | salesshare | table       | public.tickit_event_redshift    |
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d
 INBOUND    | salesshare | table       | public.tickit_listing_redshift  |
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d
 INBOUND    | salesshare | table       | public.tickit_sales_redshift    |
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d
 INBOUND    | salesshare | schema      | public                          |
 123456789012     | dd8772e1-d792-4fa4-996b-1870577efc0d
(8 rows)
```

2. Creare database locali che fanno riferimento alle unità di condivisione dati. Specificare NAMESPACE e ID account durante la creazione del database dall'unità di condivisione dati. Per ulteriori informazioni, consulta CREATE DATABASE.

```
CREATE DATABASE sales_db FROM DATASHARE salesshare OF ACCOUNT '123456789012'
 NAMESPACE 'dd8772e1-d792-4fa4-996b-1870577efc0d';
```

Se desideri un controllo più granulare sull'accesso agli oggetti nel database locale, utilizza la clausola WITH PERMISSIONS durante la creazione del database. In tal modo puoi assegnare le autorizzazioni per gli oggetti del database nel passaggio 4.

```
CREATE DATABASE sales_db WITH PERMISSIONS FROM DATASHARE salesshare OF ACCOUNT
 '123456789012' NAMESPACE 'dd8772e1-d792-4fa4-996b-1870577efc0d';
```

È possibile visualizzare i database creati dall'unità di condivisione dati eseguendo un query sulla vista SVV_REDSHIFT_DATABASE. Non è possibile connettersi a questi database creati da unità di condivisione dati e sono di sola lettura. Tuttavia, è possibile connettersi a un database locale nel cluster consumer ed eseguire una query tra database sui dati dei database creati dalle unità di condivisione dati. Non è possibile creare una unità di condivisione dati sugli oggetti di database creati da una unità di condivisione dati esistente. Tuttavia, è possibile copiare i dati in una tabella separata nel cluster di consumer, eseguire qualsiasi elaborazione necessaria e quindi condividere i nuovi oggetti creati.

3. (Facoltativo) Creare schemi esterni per fare riferimento e assegnare autorizzazioni granulari a schemi specifici nel database consumer importato nel cluster consumer. Per ulteriori informazioni, consulta CREATE EXTERNAL SCHEMA.

```
CREATE EXTERNAL SCHEMA sales_schema FROM REDSHIFT DATABASE 'sales_db' SCHEMA
 'public';
```

4. Assegna ai ruoli o agli utenti nel cluster consumer le autorizzazioni per i database e i riferimenti allo schema creati dalle unità di condivisione dati in base alle esigenze. Per ulteriori informazioni, consulta GRANT o REVOKE.

```
GRANT USAGE ON DATABASE sales_db TO Bob;
```

```
GRANT USAGE ON SCHEMA sales_schema TO ROLE Analyst_role;
```

Se hai creato il database senza la clausola WITH PERMISSIONS, puoi assegnare agli utenti o ai ruoli solo le autorizzazioni per l'intero database creato dall'unità di condivisione dati. In alcuni casi, sono necessari controlli a grana fine su un sottoinsieme di oggetti di database creati dall'unità di condivisione dati. In tal caso, è possibile creare un riferimento allo schema esterno che punta a schemi specifici nell'unità di condivisione dati come descritto nel passaggio precedente. È quindi possibile fornire autorizzazioni granulari a livello di schema. È inoltre possibile creare viste ad

associazione tardiva sugli oggetti condivisi e utilizzarle per assegnare autorizzazioni granulari. È inoltre possibile considerare che i cluster producer creino ulteriori unità di condivisione dati con la granularità richiesta. Puoi definire tutti i riferimenti allo schema di cui hai bisogno per il database creato dall'unità di condivisione dati.

Se hai creato il database con la clausola WITH PERMISSIONS nel passaggio 2, devi assegnare le autorizzazioni per gli oggetti nel database condiviso. Un utente con solo l'autorizzazione USAGE non può accedere a nessun oggetto in un database creato con la clausola WITH PERMISSIONS finché non ottiene le autorizzazioni aggiuntive a livello di oggetto.

```
GRANT SELECT ON sales_db.public.tickit_sales_redshift to Bob;
```

5. Eseguire una query sui dati negli oggetti condivisi nelle unità di condivisione dati.

   Gli utenti e i ruoli con autorizzazioni per i database consumer e gli schemi nei cluster consumer possono esplorare e navigare tra i metadati di qualsiasi oggetto condiviso. Possono inoltre esplorare e navigare tra oggetti locali in un cluster consumer. A tale scopo, utilizzare i driver JDBC o ODBC o le viste SVV_ALL e SVV_REDSHIFT.

   I cluster producer potrebbero avere molti schemi nel database, nelle tabelle e nelle viste all'interno di ogni schema. Gli utenti sul lato consumer possono vedere solo il sottoinsieme di oggetti che sono resi disponibili attraverso l'unità di condivisione dati. Questi utenti non possono visualizzare tutti i metadati dal cluster producer. Questo approccio consente di fornire un controllo granulare della sicurezza dei metadati con la condivisione dei dati.

   È possibile continuare a connettersi ai database locali nel cluster. Ma ora, è anche possibile leggere dai database e dagli schemi creati dall'unità di condivisione dati utilizzando la notazione database.schema.tabella in tre parti. È possibile eseguire query che si estendono su tutti i database visibili. Questi possono essere database locali sul cluster o database creati dalle unità di condivisione dati. I cluster consumer non possono connettersi ai database creati dalle unità di condivisione dati.

   È possibile accedere ai dati mediante la qualifica completa. Per ulteriori informazioni, consulta Esempi di utilizzo di una query tra database.

```
SELECT * FROM sales_db.public.tickit_sales_redshift;
```

È possibile utilizzare le istruzioni SELECT solo su oggetti condivisi. Tuttavia, è possibile creare tabelle nel cluster consumer eseguendo una query sui dati degli oggetti condivisi in un database locale diverso.

Oltre ad eseguire query, i consumer possono creare viste su oggetti condivisi. Sono supportate solo le viste ad associazione tardiva o le viste materializzate. Amazon Redshift non supporta le viste regolari sui dati condivisi. Le viste create dai consumer possono estendersi su più database locali o database creati dalle unità di condivisione dati. Per ulteriori informazioni, consulta CREATE VIEW.

```
// Connect to a local cluster database

// Create a view on shared objects and access it.
CREATE VIEW sales_data
AS SELECT *
FROM sales_db.public.tickit_sales_redshift
WITH NO SCHEMA BINDING;

SELECT * FROM sales_data;
```

Condivisione dei dati tra Regioni AWS

È possibile condividere i dati a scopo di lettura tra diversi cluster Amazon Redshift in Regioni AWS. Con la condivisione dei dati tra regioni, puoi condividere i dati Regioni AWS senza la necessità di copiarli manualmente. Non è necessario scaricare i dati in Amazon S3 e copiare i dati in un nuovo cluster Amazon Redshift o eseguire copie snapshot tra regioni.

Con la condivisione dei dati tra regioni, puoi condividere i dati tra cluster nello stesso Account AWS o in diversi Account AWS anche quando i cluster si trovano in regioni diverse. Quando condividi dati con cluster Amazon Redshift uguali Account AWS ma diversi Regioni AWS, segui lo stesso flusso di lavoro della condivisione dei dati all'interno di un. Account AWS Per ulteriori informazioni, consulta Condivisione dell'accesso in lettura ai dati all'interno di un Account AWS.

Se i cluster che condividono dati si trovano in aree diverse Account AWS Regioni AWS, puoi seguire lo stesso flusso di lavoro utilizzato per la condivisione dei dati Account AWS e includere associazioni a livello regionale nel cluster di consumatori. La condivisione dei dati tra regioni supporta l'associazione di datashare con i namespace del cluster interi Account AWS Regione AWS, interi o

specifici all'interno di un. Regione AWS Per ulteriori informazioni sulla condivisione dei dati, consulta. Account AWSCondivisione dei dati tra Account AWS

Quando utilizza dati da una regione diversa, il consumer paga la tariffa per il trasferimento di dati tra regioni dalla regione producer alla regione consumer.

Per utilizzare l'unità di condivisione dati, un amministratore di account consumer può associare l'unità di condivisione dati in uno dei tre modi seguenti.

- Associazione con un insieme che Account AWS abbraccia tutti i suoi Regioni AWS
- Associazione con uno specifico Regione AWS in un Account AWS
- Associazione con namespace di cluster specifici all'interno di un Regione AWS

Quando l'amministratore sceglie l'intero Account AWS, tutti i namespace del cluster esistenti e futuri Regioni AWS in diversi punti dell'account hanno accesso alle condivisioni di dati. Un amministratore di account consumer può anche scegliere namespace specifici Regioni AWS o cluster all'interno di una regione per concedere loro l'accesso alle condivisioni di dati.

Se sei un amministratore del cluster di produttori o un proprietario del database, creare un datashare, aggiungere oggetti di database e consumatori di dati al datashare e concedere autorizzazioni ai consumatori di dati. Per ulteriori informazioni, consulta Operazioni dell'amministratore del cluster producer.

Se sei un amministratore di account produttore, autorizza le condivisioni di dati utilizzando AWS Command Line Interface (AWS CLI) o la console Amazon Redshift e scegli i consumatori di dati.

Se si è un amministratore di account consumer, completare la seguente procedura:

Per associare una o più condivisioni di dati condivise da altri account ai tuoi namespace interi Account AWS o specifici Regioni AWS o a cluster all'interno di un account, usa Regione AWS la console Amazon Redshift.

Con la condivisione dei dati tra regioni, puoi aggiungere cluster in uno specifico Regione AWS utilizzando la console AWS Command Line Interface () o AWS CLI Amazon Redshift.

Per specificare una o più AWS regioni, è possibile utilizzare il comando `associate-data-share-consumer` CLI con l'opzione opzionale`consumer-region`.

Con la CLI, l'esempio seguente associa l'`Salesshare`intero all'opzione Account AWS . `associate-entire-account` È possibile associare una sola regione alla volta.

```
aws redshift associate-data-share-consumer
--region {PRODUCER_REGION}
--data-share-arn arn:aws:redshift:{PRODUCER_REGION}:{PRODUCER_ACCOUNT}:datashare:
{PRODUCER_CLUSTER_NAMESPACE}/Salesshare
--associate-entire-account
```

L'esempio seguente associa `Salesshare` con la regione Stati Uniti orientali (Ohio) (`us-east-2`).

```
aws redshift associate-data-share-consumer
--region {PRODUCER_REGION}
--data-share-arn arn:aws:redshift:{PRODUCER_REGION}:0123456789012:datashare:
{PRODUCER_CLUSTER_NAMESPACE}/Salesshare
--consumer-region 'us-east-2'
```

L'esempio seguente associa a uno spazio dei `Salesshare` nomi di cluster di consumatori specifico in un altro Account AWS spazio nella regione Asia Pacifico (Sydney) (). `ap-southeast-2`

```
aws redshift associate-data-share-consumer
--data-share-arn arn:aws:redshift:{PRODUCER_REGION}:{PRODUCER_ACCOUNT}:datashare:
{PRODUCER_CLUSTER_NAMESPACE}/Salesshare
--consumer-arn 'arn:aws:redshift:ap-southeast-2:{CONSUMER_ACCOUNT}:namespace:
{ConsumerImmutableClusterId}'
```

Puoi utilizzare la console Amazon Redshift per associare le condivisioni di dati ai tuoi namespace interi Account AWS o specifici Regioni AWS o ai cluster all'interno di un. Regione AWSA tale scopo, accedi a https://console.aws.amazon.com/redshiftv2/. Quindi associa uno o più unità di condivisione dati condivisi da altri account con l'intero Account AWS, l'intero Regione AWS o uno spazio dei nomi cluster specifico all'interno di una Regione AWS. Per ulteriori informazioni, consulta Associazione delle unità di condivisione dati.

Dopo aver associato i namespace Account AWS o quelli specifici del cluster, i datashare diventano disponibili per l'utilizzo. L'associazione dell'unità di condivisione dati può essere modificata in qualsiasi momento. Quando si modifica l'associazione da un singolo namespace di cluster a uno, Amazon Account AWS Redshift sovrascrive i namespace del cluster con le informazioni. Account AWS Quando si modifica l'associazione da uno spazio dei nomi di cluster Account AWS a uno specifico, Amazon Redshift sovrascrive le informazioni con le Account AWS informazioni sullo spazio dei nomi del cluster. Quando si modifica l'associazione da un'intera regione Account AWS a specifici namespace di cluster e AWS regioni, Amazon Redshift sovrascrive le Account AWS informazioni con le informazioni specifiche relative alla regione e allo spazio dei nomi del cluster.

Se sei un amministratore del cluster consumer puoi creare database locali che fanno riferimento alle unità di condivisione dati e assegnare le autorizzazioni per i database creati dalle unità di condivisione dati agli utenti o ai ruoli nel cluster consumer in base alle esigenze. Inoltre, è possibile creare viste su oggetti condivisi e schemi esterni per fare riferimento e assegnare autorizzazioni granulari a schemi specifici nel database consumer importato nel cluster consumer. Per ulteriori informazioni, consulta Operazioni dell'amministratore del cluster consumer.

Gestione del controllo dei costi per la condivisione dei dati tra regioni

Quando utilizza dati da una regione diversa, il consumer paga la tariffa per il trasferimento di dati tra regioni dalla regione producer alla regione consumer. Il prezzo del trasferimento dei dati è diverso per le diverse regioni. L'addebito si basa sui byte di dati analizzati per ogni esecuzione di query riuscita. Per ulteriori informazioni sui prezzi di Amazon Redshift, consultare Prezzi di Amazon Redshift.

Ti verrà addebitato un importo per il numero di byte, arrotondato al megabyte successivo, con un minimo di 10 MB per query. Puoi impostare i controlli dei costi sull'utilizzo della query e visualizzare la quantità di dati trasferiti per query sul cluster.

Per monitorare e controllare l'utilizzo e i costi associati dell'utilizzo della condivisione dei dati tra regioni, puoi creare limiti di utilizzo giornalieri, settimanali e mensili e definire le operazioni eseguite automaticamente da Amazon Redshift se tali limiti vengono raggiunti per mantenere il budget con la prevedibilità. Per ulteriori informazioni sui limiti di utilizzo in Amazon Redshift, consulta Gestione dei limiti di utilizzo in Amazon Redshift.

A seconda dei limiti di utilizzo impostati, le azioni intraprese da Amazon Redshift possono essere la registrazione di un evento su una tabella di sistema, l'invio di un CloudWatch allarme e una notifica a un amministratore con Amazon SNS o la disattivazione della condivisione dei dati tra regioni per un ulteriore utilizzo. Per ulteriori informazioni sulle operazioni, consulta Gestione dei limiti di utilizzo in Amazon Redshift.

Per definire un limite nella console Amazon Redshift, scegli Configura limite di utilizzo in Operazioni per il cluster. Puoi monitorare le tendenze di utilizzo e ricevere avvisi sull'utilizzo che supera i limiti definiti con CloudWatch metriche generate automaticamente dalle schede Prestazioni del cluster o Monitoraggio. Puoi creare, modificare ed eliminare i limiti di utilizzo a livello di programmazione tramite la AWS CLI o le operazioni API di Amazon Redshift. Per ulteriori informazioni, consulta Gestione dei limiti di utilizzo in Amazon Redshift.

Condivisione di dati con licenza Amazon Redshift su AWS Data Exchange

Quando creano AWS Data Exchange condivisioni di dati e le aggiungono a un AWS Data Exchange prodotto, i fornitori possono concedere in licenza i dati in Amazon Redshift per consentire ai consumatori di scoprire, abbonarsi e up-to-date interrogare i dati in Amazon Redshift quando dispongono di abbonamenti attivi. AWS Data Exchange

Con le AWS Data Exchange condivisioni di dati aggiunte a un AWS Data Exchange prodotto, i consumatori hanno automaticamente accesso alle condivisioni di dati del prodotto all'inizio dell'abbonamento e mantengono l'accesso finché l'abbonamento è attivo.

Lavorare con i datashare in qualità di produttore AWS Data Exchange

Se sei un amministratore di un cluster di produttori, segui questi passaggi per gestire le AWS Data Exchange condivisioni di dati sulla console Amazon Redshift:

1. Crea datashare nel tuo cluster su cui condividere dati AWS Data Exchange e concedere l'accesso ai datashare. AWS Data Exchange

   Le unità di condivisione dati possono essere create dall'utente con privilegi avanzati e dai proprietari di database del cluster. Ogni unità di condivisione dati è associata a un database durante la creazione. Solo gli oggetti di quel database possono essere condivisi in quella unità di condivisione dati. Sullo stesso database possono essere create più unità di condivisione dati con la stessa granularità di oggetti o con una granularità differente. Non vi è alcun limite sul numero di unità di condivisione dati che un cluster può creare.

   Per creare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Creazione di unità di condivisione dati.

   Utilizza l'opzione MANAGEDBY ADX per concedere implicitamente l'accesso al datashare a durante l'esecuzione dell'istruzione CREATE DATASHARE. AWS Data Exchange Ciò AWS Data Exchange indica che gestisce questo datashare. Puoi utilizzare l'opzione MANAGEDBY ADX solo quando crei una nuova unità di condivisione dati. Non è possibile utilizzare l'istruzione ALTER DATASHARE per modificare un'unità di condivisione dati esistente per aggiungere l'opzione MANAGEDBY ADX. Una volta creato una unità di condivisione dati con l'opzione MANAGEDBY ADX, solo AWS Data Exchange può accedere e gestire l'unità di condivisione dati.

   ```
   CREATE DATASHARE salesshare
   ```

```
[[SET] MANAGEDBY [=] {ADX} ];
```

2. Aggiungere oggetti alle unità di condivisione dati. L'amministratore di Producer continua a gestire gli oggetti datashare disponibili in un datashare. AWS Data Exchange

   Per aggiungere oggetti a una unità di condivisione dati, aggiungere lo schema prima di aggiungere gli oggetti. Quando si aggiunge uno schema, Amazon Redshift non aggiunge tutti gli oggetti. È necessario aggiungerli esplicitamente. Per ulteriori informazioni, consulta ALTER DATASHARE.

   ```
   ALTER DATASHARE salesshare ADD SCHEMA PUBLIC;
   ALTER DATASHARE salesshare ADD TABLE public.tickit_sales_redshift;
   ALTER DATASHARE salesshare ADD ALL TABLES IN SCHEMA PUBLIC;
   ```

   A una unità di condivisione dati è possibile aggiungere anche le viste.

   ```
   CREATE VIEW public.sales_data_summary_view AS SELECT * FROM
    public.tickit_sales_redshift;
   ALTER DATASHARE salesshare ADD TABLE public.sales_data_summary_view;
   ```

   Utilizzare ALTER DATASHARE per condividere schemi e tabelle, viste e funzioni in un determinato schema. Gli utenti con privilegi avanzati, i proprietari di unità di condivisione dati o gli utenti che dispongono dell'autorizzazione ALTER o ALL sull'unità di condivisione dati possono modificarla in modo da aggiungere o rimuovere oggetti. Gli utenti devono disporre delle autorizzazioni per aggiungere o rimuovere oggetti dall'unità di condivisione dati. Gli utenti devono inoltre essere i proprietari degli oggetti o disporre delle autorizzazioni SELECT, USAGE o ALL sugli oggetti.

   Utilizzare la clausola INCLUDENEW per aggiungere nuove tabelle, viste o funzioni definite dall'utente (FDU) SQL future create in uno scherma specificato nell'unità di condivisione dati. Solo gli utenti con privilegi avanzati possono modificare questa proprietà per ogni coppia unità di condivisione dati-schema.

   ```
   ALTER DATASHARE salesshare ADD SCHEMA PUBLIC;
   ALTER DATASHARE salesshare SET INCLUDENEW = TRUE FOR SCHEMA PUBLIC;
   ```

   Per aggiungere o rimuovere oggetti dalle unità di condivisione dati è possibile utilizzare la console Amazon Redshift. Per ulteriori informazioni, consultare Aggiunta di oggetti di unità di

condivisione dati alle unità di condivisione dati, Rimozione di oggetti di unità di condivisione dati dalle unità di condivisione dati e Modifica delle condivisioni AWS Data Exchange di dati.

3. Per autorizzare l'accesso ai datashare per, effettuate una delle seguenti operazioni: AWS Data Exchange

   - Autorizza esplicitamente l'accesso al datashare utilizzando la parola chiave nell'API AWS Data Exchange . ADX `aws redshift authorize-data-share` Ciò consente di riconoscere il AWS Data Exchange datashare nell'account del servizio e gestire l'associazione dei consumatori al datashare.

     ```
     aws redshift authorize-data-share
     --data-share-arn arn:aws:redshift:us-east-1:{PRODUCER_ACCOUNT}:datashare:
     {PRODUCER_CLUSTER_NAMESPACE}/salesshare
     --consumer-identifier ADX
     ```

     È possibile utilizzare una chiave condizionale `ConsumerIdentifier` per `AuthorizeDataShare` e API `DeauthorizeDataShare` per consentire o negare esplicitamente a AWS Data Exchange ad effettuare chiamate alle due API nella policy IAM.

     ```
     {
         "Version": "2012-10-17",
         "Statement": [
             {
                 "Sid": "VisualEditor0",
                 "Effect": "Deny",
                 "Action": [
                     "redshift:AuthorizeDataShare",
                     "redshift:DeauthorizeDataShare"
                 ],
                 "Resource": "*",
                 "Condition": {
                     "StringEqualsIgnoreCase": {
                         "redshift:ConsumerIdentifier": "ADX"
                     }
                 }
             }
         ]
     }
     ```

- Utilizza la console Amazon Redshift per autorizzare o rimuovere l'autorizzazione delle condivisioni di dati. AWS Data Exchange Per ulteriori informazioni, consulta Autorizzazione o rimozione dell'autorizzazione dalle unità di condivisione dati.
- Facoltativamente, puoi autorizzare implicitamente l'accesso al datashare quando importi il AWS Data Exchange datashare in un set di dati. AWS Data Exchange

Per rimuovere l'autorizzazione all'accesso ai AWS Data Exchange datashare, utilizza la parola chiave nell'operazione API. ADX `aws redshift deauthorize-data-share` In questo modo, consenti a AWS Data Exchange di riconoscere l'unità nell'account del servizio e gestire la rimozione dell'associazione dall'unità.

```
aws redshift deauthorize-data-share
--data-share-arn arn:aws:redshift:us-east-1:{PRODUCER_ACCOUNT}:datashare:
{PRODUCER_CLUSTER_NAMESPACE}/salesshare
--consumer-identifier ADX
```

4. Elencare le unità di condivisione dati create nel cluster ed esaminare il contenuto dell'unità.

Nell'esempio seguente sono riportate le informazioni di una unità di condivisione dati denominata salesshare. Per ulteriori informazioni, consulta DESC DATASHARE e SHOW DATASHARES.

```
DESC DATASHARE salesshare;

 producer_account  |          producer_namespace          | share_type | share_name
 | object_type |           object_name          |   include_new
-------------------+--------------------------------------+------------
+------------+-------------+---------------------------------+-------------------
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_users_redshift   |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_venue_redshift   |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_category_redshift|
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_date_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_event_redshift   |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_listing_redshift |
```

```
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | table       | public.tickit_sales_redshift   |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | schema      | public                         |  t
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | OUTBOUND   | salesshare
 | view        | public.sales_data_summary_view |
```

L'esempio seguente mostra le unità di condivisione dati in uscita in un cluster producer.

```
SHOW DATASHARES LIKE 'sales%';
```

L'output è simile al seguente.

```
share_name | share_owner  | source_database | consumer_database | share_type |
 createdate      | is_publicaccessible  | share_acl | producer_account |
 producer_namespace
-----------+--------------+-----------------+-------------------+------------
+---------------------+----------------------+-----------+-----------------
+--------------------------------------
salesshare |    100       | dev             |                   |  OUTBOUND
 | 2020-12-09 02:27:08 |          True        |           |   123456789012   |
 13b8833d-17c6-4f16-8fe4-1a018f5ed00d
```

Per ulteriori informazioni, consulta DESC DATASHARE e SHOW DATASHARES.

È possibile utilizzare anche SVV_DATASHARES, SVV_DATASHARE_CONSUMERS e SVV_DATASHARE_OBJECTS per visualizzare le unità di condivisione dati, gli oggetti all'interno dell'unità di condivisione dati e i consumer dell'unità di condivisione dati.

5. Eliminare le unità di condivisione dati. Ti consigliamo di non eliminare un AWS Data Exchange datashare condiviso con altri Account AWS utilizzando l'istruzione DROP DATASHARE. Tali account perderanno l'accesso all'unità di condivisione dati. Questa operazione è irreversibile. Ciò potrebbe violare i termini dell'offerta del prodotto Data in. AWS Data Exchange Se desideri eliminare un AWS Data Exchange datashare, consulta. Note per l'utilizzo di DROP DATASHARE

   Nell'esempio seguente viene rimossa una unità di condivisione dati denominata salesshare.

```
DROP DATASHARE salesshare;
ERROR:  Drop of ADX-managed datashare salesshare requires session variable
 datashare_break_glass_session_var to be set to value '620c871f890c49'
```

Per consentire l'eliminazione di un AWS Data Exchange datashare, imposta la variabile datashare_break_glass_session_var ed esegui nuovamente l'istruzione DROP DATASHARE. Se desideri eliminare un datashare, Note per l'utilizzo di DROP DATASHARE consulta. AWS Data Exchange

Per eliminare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Eliminazione delle condivisioni di AWS Data Exchange dati create nell'account.

6. Utilizzare ALTER DATASHARE per rimuovere gli oggetti dalle unità di condivisione dati in qualsiasi punto dall'unità. Utilizzare REVOKE USAGE ON per revocare le autorizzazioni sull'unità di condivisione dati a su consumer. Revoca le autorizzazioni USAGE sugli oggetti all'interno di una unità di condivisione dati e disabilita immediatamente l'accesso a tutti i cluster di consumer. L'elenco delle unità di condivisione dati e delle query dei metadati, ad esempio l'elenco dei database e delle tabelle, non restituisce gli oggetti condivisi dopo la revoca dell'accesso.

```
ALTER DATASHARE salesshare REMOVE TABLE public.tickit_sales_redshift;
```

Per modificare le unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Modifica delle condivisioni AWS Data Exchange di dati.

7. Concedi o revoca GRANT USAGE dai datashare. AWS Data Exchange Non è possibile concedere o revocare GRANT USAGE per il datashare. AWS Data Exchange L'esempio seguente mostra un errore quando l'autorizzazione GRANT USAGE viene concessa a un datashare che Account AWS gestisce. AWS Data Exchange

```
CREATE DATASHARE salesshare MANAGEDBY ADX;
```

```
GRANT USAGE ON DATASHARE salesshare TO ACCOUNT '012345678910';
ERROR:  Permission denied to add/remove consumer to/from datashare salesshare.
 Datashare consumers are managed by ADX.
```

Per ulteriori informazioni, consulta GRANT o REVOKE.

Se sei un amministratore di un cluster di produttori, segui questi passaggi per creare e pubblicare un prodotto datashare sulla console: AWS Data Exchange

- Una volta creato il AWS Data Exchange datashare, il produttore crea un nuovo set di dati, importa risorse, crea una revisione e crea e pubblica un nuovo prodotto.

  Utilizzare la console Amazon Redshift per creare set di dati. Per ulteriori informazioni, consulta Creazione di set di dati su AWS Data Exchange.

  Per ulteriori informazioni, consulta Fornitura di prodotti di dati su. AWS Data Exchange

Utilizzo delle condivisioni AWS Data Exchange di dati in qualità di consumatore

Se sei un consumatore, segui questi passaggi per scoprire prodotti di dati che contengono AWS Data Exchange condivisioni di dati e interrogare i dati di Amazon Redshift:

1. Sulla AWS Data Exchange console, scopri e sottoscrivi prodotti di dati che contengono datashare. AWS Data Exchange

   Una volta avviato l'abbonamento, puoi accedere ai dati con licenza di Amazon Redshift importati come asset in set di dati che contengono condivisioni di dati. AWS Data Exchange

   Per ulteriori informazioni su come iniziare a utilizzare prodotti di dati che contengono AWS Data Exchange datashare, consulta Abbonamento a prodotti di dati su. AWS Data Exchange
2. Sulla console Amazon Redshift, crea un cluster Amazon Redshift, se necessario.

   Per informazioni su come creare un cluster, consultare Creazione di un cluster.
3. Elencare le unità di condivisione dati rese disponibili e visualizzare il contenuto delle unità di condivisione dati. Per ulteriori informazioni, consulta DESC DATASHARE e SHOW DATASHARES.

   Nell'esempio seguente vengono visualizzate le informazioni relative alle unità di condivisione dati in ingresso di uno spazio dei nomi producer specificato. Quando si esegue DESC DATASHARE come amministratore del cluster consumer, è necessario specificare l'opzione ACCOUNT e NAMESPACE per visualizzare le unità di condivisione dati in ingresso.

```
DESC DATASHARE salesshare of ACCOUNT '123456789012' NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';

 producer_account  |          producer_namespace          | share_type | share_name
 | object_type |           object_name           |   include_new
-------------------+--------------------------------------+------------
+------------+-------------+----------------------------------+------------------
```

```
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_users_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_venue_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_category_redshift |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_date_redshift     |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_event_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_listing_redshift  |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | table       | public.tickit_sales_redshift    |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | schema      | public                          |
 123456789012      | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare
 | view        | public.sales_data_summary_view  |
```

Solo gli utenti con privilegi avanzati per il cluster possono completare questa opzione. È possibile utilizzare anche SVV_DATASHARES per visualizzare le unità di condivisione dati e SVV_DATASHARE_OBJECTS per visualizzare gli oggetti all'interno dell'unità di condivisione dati.

L'esempio seguente mostra le unità di condivisione dati in uscita in un cluster producer.

```
SHOW DATASHARES LIKE 'sales%';


 share_name | share_owner | source_database | consumer_database | share_type
 | createdate | is_publicaccessible | share_acl | producer_account |
 producer_namespace
------------+-------------+-----------------+-------------------+------------
+------------+---------------------+-----------+------------------
+--------------------------------------
 salesshare |             |                 |                   | INBOUND
    |            |         t           |           |   123456789012   |
 13b8833d-17c6-4f16-8fe4-1a018f5ed00d
```

4. Creare database locali che fanno riferimento alle unità di condivisione dati. È necessario specificare le opzioni ACCOUNT e NAMESPACE per creare database locali per le condivisioni di dati. AWS Data Exchange Per ulteriori informazioni, consulta CREATE DATABASE.

```
CREATE DATABASE sales_db FROM DATASHARE salesshare OF ACCOUNT '123456789012'
 NAMESPACE '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

Se desideri un controllo più granulare sull'accesso agli oggetti nel database locale, utilizza la clausola WITH PERMISSIONS durante la creazione del database. In tal modo puoi assegnare le autorizzazioni per gli oggetti del database nel passaggio 6.

```
CREATE DATABASE sales_db WITH PERMISSIONS FROM DATASHARE salesshare OF ACCOUNT
 '123456789012' NAMESPACE '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

È possibile vedere i database creati dall'unità di condivisione dati eseguendo una query sulla vista SVV_REDSHIFT_DATABASE. Non è possibile connettersi a questi database creati da unità di condivisione dati e sono di sola lettura. Tuttavia, è possibile connettersi a un database locale nel cluster consumer ed eseguire una query tra database sui dati dei database creati dalle unità di condivisione dati. Non è possibile creare una unità di condivisione dati sugli oggetti di database creati da una unità di condivisione dati esistente. Tuttavia, è possibile copiare i dati in una tabella separata nel cluster consumer, eseguire qualsiasi elaborazione necessaria e quindi condividere i nuovi oggetti creati.

Per creare i database dalle unità di condivisione dati è possibile utilizzare anche la console Amazon Redshift. Per ulteriori informazioni, consulta Creazione di database da unità di condivisione dati.

5. (Facoltativo) Creare schemi esterni per fare riferimento e assegnare autorizzazioni granulari a schemi specifici nel database consumer importato nel cluster consumer. Per ulteriori informazioni, consulta CREATE EXTERNAL SCHEMA.

```
CREATE EXTERNAL SCHEMA sales_schema FROM REDSHIFT DATABASE 'sales_db' SCHEMA
 'public';
```

6. Assegna ai ruoli o agli utenti nel cluster consumer le autorizzazioni per i database e i riferimenti allo schema creati dalle unità di condivisione dati in base alle esigenze. Per ulteriori informazioni, consulta GRANT o REVOKE.

```
GRANT USAGE ON DATABASE sales_db TO Bob;
```

```
GRANT USAGE ON SCHEMA sales_schema TO ROLE Analyst_role;
```

Se hai creato il database senza la clausola WITH PERMISSIONS, puoi assegnare agli utenti e ai ruoli solo le autorizzazioni per l'intero database creato dall'unità di condivisione dati. In alcuni casi, sono necessari controlli a grana fine su un sottoinsieme di oggetti di database creati dall'unità di condivisione dati. In tal caso, è possibile creare un riferimento allo schema esterno che punti a schemi specifici nell'unità di condivisione dati (come descritto nel passaggio precedente) e fornire autorizzazioni granulari a livello di schema.

È inoltre possibile creare viste ad associazione tardiva sugli oggetti condivisi e utilizzarle per assegnare autorizzazioni granulari. È inoltre possibile considerare che i cluster producer creino ulteriori unità di condivisione dati con la granularità richiesta. Puoi definire tutti i riferimenti allo schema di cui hai bisogno per il database creato dall'unità di condivisione dati.

Se hai creato il database con la clausola WITH PERMISSIONS nel passaggio 4, devi assegnare le autorizzazioni per gli oggetti nel database condiviso. Un utente con solo l'autorizzazione USAGE non può accedere a nessun oggetto in un database creato con la clausola WITH PERMISSIONS finché non ottiene le autorizzazioni aggiuntive a livello di oggetto.

```
GRANT SELECT ON sales_db.public.tickit_sales_redshift to Bob;
```

7. Eseguire una query sui dati negli oggetti condivisi nelle unità di condivisione dati.

   Gli utenti e i ruoli con autorizzazioni per i database consumer e gli schemi nei cluster consumer possono esplorare e navigare tra i metadati di qualsiasi oggetto condiviso. Possono inoltre esplorare e navigare tra oggetti locali in un cluster consumer. Per fare ciò, utilizzano i driver JDBC o ODBC o le viste SVV_ALL e SVV_REDSHIFT.

   I cluster producer potrebbero avere molti schemi nel database, nelle tabelle e nelle viste all'interno di ogni schema. Gli utenti sul lato consumer possono vedere solo il sottoinsieme di oggetti che sono resi disponibili attraverso l'unità di condivisione dati. Questi utenti non possono visualizzare gli interi metadati dal cluster producer. Questo approccio consente di fornire un controllo granulare della sicurezza dei metadati con la condivisione dei dati.

   È possibile continuare a connettersi ai database locali nel cluster. Ma ora, è anche possibile leggere dai database e dagli schemi creati dall'unità di condivisione dati utilizzando la notazione database.schema.tabella in tre parti. È possibile eseguire query che si estendono su tutti i database visibili. Questi possono essere database locali sul cluster o database creati dalle unità di condivisione dati. I cluster consumer non possono connettersi ai database creati dalle unità di condivisione dati.

È possibile accedere ai dati mediante la qualifica completa. Per ulteriori informazioni, consulta Esempi di utilizzo di una query tra database.

```
SELECT * FROM sales_db.public.tickit_sales_redshift ORDER BY 1,2 LIMIT 5;

 salesid | listid | sellerid | buyerid | eventid | dateid | qtysold | pricepaid |
 commission |      saletime
---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------
       1 |      1 |    36861 |   21191 |    7872 |   1875 |       4 |    728.00 |
   109.20 | 2008-02-18 02:36:48
       2 |      4 |     8117 |   11498 |    4337 |   1983 |       2 |     76.00 |
    11.40 | 2008-06-06 05:00:16
       3 |      5 |     1616 |   17433 |    8647 |   1983 |       2 |    350.00 |
    52.50 | 2008-06-06 08:26:17
       4 |      5 |     1616 |   19715 |    8647 |   1986 |       1 |    175.00 |
    26.25 | 2008-06-09 08:38:52
       5 |      6 |    47402 |   14115 |    8240 |   2069 |       2 |    154.00 |
    23.10 | 2008-08-31 09:17:02
```

È possibile utilizzare le istruzioni SELECT solo su oggetti condivisi. Tuttavia, è possibile creare tabelle nel cluster consumer eseguendo una query sui dati degli oggetti condivisi in un database locale diverso.

Oltre alle query, i consumer possono creare viste su oggetti condivisi. Sono supportate solo le viste ad associazione tardiva o le viste materializzate. Amazon Redshift non supporta le viste regolari sui dati condivisi. Le viste create dai consumer possono estendersi su più database locali o database creati dalle unità di condivisione dati. Per ulteriori informazioni, consulta CREATE VIEW.

```
// Connect to a local cluster database

// Create a view on shared objects and access it.
CREATE VIEW sales_data
AS SELECT *
FROM sales_db.public.tickit_sales_redshift
WITH NO SCHEMA BINDING;

SELECT * FROM sales_data;
```

Lavorare con -managed datashares AWS Lake Formation

La condivisione dei dati AWS Lake Formation consente di definire centralmente AWS Lake Formation le autorizzazioni delle condivisioni di dati Amazon Redshift e limitare l'accesso degli utenti agli oggetti all'interno di un datashare.

Utilizzo di unità di condivisione dati gestite da Lake Formation in qualità di producer

Se sei l'amministratore di un gruppo di lavoro o un cluster producer, segui questa procedura per condividere le unità di condivisione dati con Lake Formation:

1. Crea datashare nel tuo cluster e autorizza l'accesso ai datashare. AWS Lake Formation

   Le unità di condivisione dati possono essere create solo dall'utente con privilegi avanzati e dai proprietari di database del cluster. Ogni unità di condivisione dati è associata a un database durante la creazione. Solo gli oggetti di quel database possono essere condivisi in quella unità di condivisione dati. Sullo stesso database possono essere create più unità di condivisione dati con la stessa granularità di oggetti o con una granularità differente. Non vi è alcun limite sul numero di unità di condivisione dati che un cluster può creare.

   ```
   CREATE DATASHARE salesshare;
   ```

2. Aggiungi oggetti alle unità di condivisione dati. L'amministratore di un gruppo di lavoro o un cluster producer continua a gestire gli oggetti delle unità di condivisione dati disponibili. Per aggiungere oggetti a una unità di condivisione dati, aggiungere lo schema prima di aggiungere gli oggetti. Quando si aggiunge uno schema, Amazon Redshift non aggiunge tutti gli oggetti. È necessario aggiungerli esplicitamente. Per ulteriori informazioni, consulta ALTER DATASHARE.

   ```
   ALTER DATASHARE salesshare ADD SCHEMA PUBLIC;
   ALTER DATASHARE salesshare ADD TABLE public.tickit_sales_redshift;
   ALTER DATASHARE salesshare ADD ALL TABLES IN SCHEMA PUBLIC;
   ```

   A una unità di condivisione dati è possibile aggiungere anche le viste. Sono supportate viste standard, ad associazione tardiva e materializzate.

   ```
   CREATE VIEW public.sales_data_summary_view AS SELECT * FROM
    public.tickit_sales_redshift;
   ALTER DATASHARE salesshare ADD TABLE public.tickit_sales_redshift;
   ```

Utilizza ALTER DATASHARE per condividere schemi, tabelle e viste in un determinato schema. Gli utenti con privilegi avanzati, i proprietari di unità di condivisione dati o gli utenti che dispongono dell'autorizzazione ALTER o ALL sull'unità di condivisione dati possono modificarla in modo da aggiungere o rimuovere oggetti. Gli utenti di database devono inoltre essere i proprietari degli oggetti o disporre delle autorizzazioni SELECT, USAGE o ALL sugli oggetti.

Utilizza la clausola INCLUDENEW per aggiungere nuove tabelle e viste create in uno scherma specificato nell'unità di condivisione dati. Solo gli utenti con privilegi avanzati possono modificare questa proprietà per ogni coppia unità di condivisione dati-schema.

```
ALTER DATASHARE salesshare ADD SCHEMA PUBLIC;
ALTER DATASHARE salesshare SET INCLUDENEW = TRUE FOR SCHEMA PUBLIC;
```

3. Concedi all'unità di condivisione dati l'accesso a un account amministratore di Lake Formation.

```
GRANT USAGE ON DATASHARE salesshare TO ACCOUNT '012345678910' VIA DATA CATALOG;
```

Per revocare l'autorizzazione all'utilizzo, usa il comando seguente.

```
REVOKE USAGE ON DATASHARE salesshare FROM ACCOUNT '012345678910' VIA DATA CATALOG;
```

4. Autorizza l'accesso all'unità di condivisione dati per Lake Formation utilizzando l'operazione API `aws redshift authorize-data-share`. In questo modo si consente a Lake Formation di riconoscere l'unità di condivisione dati nell'account del servizio e di gestire l'associazione dei consumer all'unità di condivisione dati.

```
aws redshift authorize-data-share
--data-share-arn arn:aws:redshift:us-east-1:{PRODUCER_ACCOUNT}:datashare:
{PRODUCER_CLUSTER_NAMESPACE}/salesshare
--consumer-identifier {"DataCatalog/<consumer-account-id>"}
```

Per rimuovere l'autorizzazione dalle unità di condivisione dati gestite da Lake Formation, utilizza l'operazione API `aws redshift deauthorize-data-share`, In questo modo, consenti di riconoscere il AWS Lake Formation datashare nell'account del servizio e di rimuovere l'autorizzazione.

```
aws redshift deauthorize-data-share
```

```
--data-share-arn arn:aws:redshift:us-east-1:{PRODUCER_ACCOUNT}:datashare:
{PRODUCER_CLUSTER_NAMESPACE}/salesshare
--consumer-identifier {"DataCatalog/<consumer-account-id>"}
```

Se in qualsiasi momento l'amministratore del gruppo di lavoro o del cluster producer decide che non è più necessario condividere i dati con il gruppo di lavoro o il cluster consumer, può utilizzare DROP DATASHARE per eliminare, annullare l'autorizzazione o revocare le autorizzazioni dell'unità di condivisione dati. Le autorizzazioni e gli oggetti associati in Lake Formation non vengono eliminati automaticamente.

```
DROP DATASHARE salesshare;
```

Dopo aver autorizzato l'account Lake Formation a gestire il datashare, l'amministratore di Lake Formation può scoprire il datashare condiviso, associare il dateshare a un ARN del catalogo dati e creare un database nel collegamento al datashare. AWS Glue Data Catalog Per AWS CLI associare associate-data-share-consumeri datashare utilizzando, utilizzare il comando. Per condividere un datashare Regioni AWS, specifica il `--region` parametro nel `associate-data-share-consumer` comando o usa la AWS console per scegliere i tuoi consumatori di dati. L'esempio seguente illustra come condividere una unità di condivisione dati gestita da Lake Formation tra le regioni.

```
aws redshift associate-data-share-consumer --region <region-1>
--data-share-arn 'arn:aws:redshift:us-
east-1:12345678912:datashare:035c45ea-61ce-86f0-8b75-19ac6102c3b7/sample_share'
--consumer-arn 'arn:aws:glue:<region-1>:111912345678:catalog'
```

L'amministratore di Lake Formation deve anche creare risorse locali che definiscano in che modo gli oggetti all'interno dell'unità di condivisione dati devono essere mappati agli oggetti all'interno di Lake Formation. Per ulteriori informazioni sul rilevamento delle unità di condivisione dati e sulla creazione di risorse locali, consulta Gestione delle autorizzazioni per i dati in una unità di condivisione dati di Amazon Redshift.

Utilizzo di unità di condivisione dati gestite da Lake Formation in qualità di consumer

Dopo AWS Glue Data Catalog che l' AWS Lake Formation amministratore ha scoperto l'invito alla condivisione di dati e creato un database collegato al datashare, l'amministratore del cluster di consumatori o del gruppo di lavoro può associare il cluster al datashare e al database in esso contenuto AWS Glue Data Catalog, creare un database locale al cluster di consumatori o al gruppo

di lavoro e concedere l'accesso a utenti e ruoli nel cluster di consumatori o nel gruppo di lavoro di Amazon Redshift per avviare le query. Segui la procedura riportata sotto per impostare le autorizzazioni per eseguire query.

1. Nella console Amazon Redshift crea un gruppo di lavoro o un cluster consumer Amazon Redshift, se necessario. Per informazioni su come creare un cluster, consulta Creazione di un cluster.
2. Per elencare a quali database del cluster di AWS Glue Data Catalog consumatori o del gruppo di lavoro hanno accesso gli utenti, esegui il comando SHOW DATABASES.

   ```
   SHOW DATABASES FROM DATA CATALOG [ACCOUNT <account-id>,<account-id2>] [LIKE
    <expression>]
   ```

   In questo modo vengono elencate le risorse disponibili nel Data Catalog, ad esempio l'ARN del AWS Glue database, il nome del database e le informazioni sul datashare.
3. Utilizzando il AWS Glue database ARN di SHOW DATABASES, crea un database locale nel cluster o nel gruppo di lavoro di consumatori. Per ulteriori informazioni, consulta la pagina CREATE DATABASE.

   ```
   CREATE DATABASE lf_db FROM ARN <lake-formation-database-ARN> WITH [NO] DATA CATALOG
    SCHEMA [<schema>];
   ```
4. Fornisci ai ruoli o agli utenti nel gruppo di lavoro o nel cluster consumer l'accesso per i database e i riferimenti allo schema creati dalle unità di condivisione dati in base alle esigenze. Per ulteriori informazioni, consulta la pagina GRANT o REVOKE. Nota: gli utenti creati utilizzando il comando CREATE USER non possono accedere agli oggetti in unità di condivisione dati che sono state condivise con Lake Formation. Solo gli utenti con accesso sia a Redshift che a Lake Formation possono accedere alle unità di condivisione dati condivise con Lake Formation.

   ```
   GRANT USAGE ON DATABASE sales_db TO IAM:Bob;
   ```

   In qualità di amministratore del gruppo di lavoro o del cluster consumer, puoi assegnare le autorizzazioni per l'intero database creato dall'unità di condivisione dati solo agli utenti e ai ruoli. In alcuni casi, sono necessari controlli a grana fine su un sottoinsieme di oggetti di database creati dall'unità di condivisione dati.

   È inoltre possibile creare viste ad associazione tardiva sugli oggetti condivisi e utilizzarle per assegnare autorizzazioni granulari. Inoltre puoi stabilire che i gruppi di lavoro o i cluster producer

creino ulteriori unità di condivisione dati con la granularità richiesta. È possibile creare il maggior numero di riferimenti allo schema al database creato dall'unità di condivisione dati.

5. Gli utenti del database possono utilizzare le viste SVV_EXTERNAL_TABLES e SVV_EXTERNAL_COLUMNAL_COLUMNS per trovare tutte le tabelle o le colonne condivise all'interno del database AWS Glue

```
SELECT * from svv_external_tables WHERE redshift_database_name = 'lf_db';

SELECT * from svv_external_columns WHERE redshift_database_name = 'lf_db';
```

6. Eseguire una query sui dati negli oggetti condivisi nelle unità di condivisione dati.

   Gli utenti e i ruoli con autorizzazioni per i database consumer e gli schemi nei gruppi di lavoro o nei cluster consumer possono esplorare e navigare tra i metadati di qualsiasi oggetto condiviso. Possono inoltre esplorare e navigare tra oggetti locali in un gruppo di lavoro o un cluster consumer. Per procedere in tal senso possono utilizzare i driver JDBC o ODBC o le viste SVV_ALL o SVV_EXTERNAL.

```
SELECT * FROM lf_db.schema.table;
```

   È possibile utilizzare le istruzioni SELECT solo su oggetti condivisi. Tuttavia, è possibile creare tabelle nel cluster consumer eseguendo una query sui dati degli oggetti condivisi in un database locale diverso.

```
// Connect to a local cluster database

// Create a view on shared objects and access it.

CREATE VIEW sales_data
AS SELECT *
FROM sales_db.public.tickit_sales_redshift
WITH NO SCHEMA BINDING;

SELECT * FROM sales_data;
```

Gestione della condivisione dei dati mediante la console

Utilizzare la console Amazon Redshift per gestire i dati creati nell'account o condivisi da altri account.

Sono necessarie le autorizzazioni per creare, modificare o eliminare le unità di condivisione dati. Per informazioni, consulta Gestione delle autorizzazioni per le unità di condivisione dati in Amazon Redshift.

- Se si è un amministratore di cluster producer, è possibile creare unità di condivisione dati, aggiungere consumer di dati, aggiungere oggetti di unità di condivisione dati, creare database dalle unità di condivisione dati, modificare unità di condivisione dati o eliminare unità di condivisione dati dalla scheda CLUSTER.

  Dal menu di navigazione, passare alla scheda Cluster e scegliere un cluster dall'elenco di cluster. Effettuare quindi una delle seguenti operazioni:

  - Seleziona la scheda Datashares (Unità di condivisione dati) e scegli un'unità di condivisione dati dalla sezione Datashares created in my namespace (Unità di condivisione dati create nel mio spazio dei nomi). Effettuare quindi una delle seguenti operazioni:
    - Creazione di unità di condivisione dati

      Quando viene creata una unità di condivisione dati, è possibile aggiungere oggetti di unità di condivisione dati o consumer di dati. Per ulteriori informazioni, consulta Aggiunta di oggetti di unità di condivisione dati alle unità di condivisione dati e Aggiunta di consumer di dati alle unità di condivisione dati.
    - Modifica delle unità di condivisione dati create nell'account
    - Eliminazione delle unità di condivisione dati create nell'account
  - Selezionare Unità di condivisione dati e scegliere una unità di condivisione dati dalla sezione Unità di condivisione dati da altri cluster. Effettuare quindi una delle seguenti operazioni:
    - Creazione di unità di condivisione dati
    - Creazione di database da unità di condivisione dati
  - Scegliere Database e selezionare un database dalla sezione Database. Quindi scegliere Crea unità di condivisione dati. Per ulteriori informazioni, consulta Creazione di database da unità di condivisione dati.

> **Note**
>
> Per visualizzare i database e gli oggetti all'interno dei database o per visualizzare le unità di condivisione dati nel cluster, connettersi a un database. Per ulteriori informazioni, consulta Connessione a un database.

Connessione a un database

Connettersi a un database per visualizzare i database e gli oggetti all'interno dei database in questo cluster o per visualizzare le unità di condivisione dati.

Le credenziali utente utilizzate per connettersi a un database specificato devono disporre delle autorizzazioni necessarie per visualizzare tutte le unità di condivisione dati.

Se non esiste una connessione locale, procedere in uno dei seguenti modi:

- Nella pagina dei dettagli del cluster, dalla tabella Databases (Database), nella sezione Databases (Database) o Datashare objects (Oggetti dell'unità di condivisione dati), scegli Connect to database (Connetti al database) per visualizzare gli oggetti di database nel cluster.
- Nella pagina dei dettagli del cluster, dalla scheda Unità di condivisione dati effettuare una delle seguenti operazioni:
  - Nella sezione Unità di condivisione dati da altri cluster, scegliere Connetti al database per visualizzare le unità di condivisione dati da altri cluster.
  - Nella sezione Unità di condivisione dati da altri cluster, scegliere Connetti al database per visualizzare le unità di condivisione dati da altri cluster.
- Nella finestra Connetti al database, completare una delle seguenti operazioni:
  - Se si sceglie Crea una nuova connessione, scegliere AWS Secrets Manager per utilizzare un segreto memorizzato per autenticare l'accesso per la connessione.

    In alternativa, scegliere Credenziali temporanee per utilizzare le credenziali del database. Specificare i valori per Nome database e Utente database.

    Scegli Connetti.
  - Scegliere Usa una connessione recente per connettersi a un altro database per cui si dispone delle autorizzazioni necessarie.

    Amazon Redshift effettua automaticamente la connessione.

Dopo aver stabilito la connessione al database, è possibile iniziare a creare unità di condivisione dati, eseguire query sulle unità di condivisione dati o creare database dalle unità di condivisione dati.

Creazione di unità di condivisione dati

Creazione di unità di condivisione dati

In qualità di amministratore del cluster producer, è possibile creare unità di condivisione dati dalle schede Database o Unità di condivisione dati nella pagina dei dettagli del cluster.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Nella pagina dei dettagli del cluster, effettuare una delle seguenti operazioni:
   - Dalla scheda Database, nella sezione Database scegliere un database. Viene visualizzata la pagina dei dettagli del database.

     Quindi scegliere Crea unità di condivisione dati. È possibile creare una unità di condivisione dati solo da un database locale. Se è la prima volta che si esegue la connessione al database, viene visualizzata la pagina Connetti al database. Seguire la procedura riportata in Connessione a un database per connettersi a un database. Se esiste una connessione recente, viene visualizzata la pagina Crea unità di condivisione dati.
   - Dalla scheda Unità di condivisione dati, nella scheda Unità di condivisione dati, connettersi a un database se non si dispone di una connessione al database.

     Nella sezione Unità di condivisione dati create nel mio cluster, scegliere Crea unità di condivisione dati. Viene visualizzata la pagina Crea unità di condivisione dati.
4. Nella sezione Informazioni su unità di condivisione dati, selezionare una delle seguenti opzioni:
   - Scegliere Unità di condivisione dati per creare unità di condivisione dati per condividere i dati a scopo di lettura su diversi cluster Amazon Redshift o nello stesso Account AWS o diversi Account AWS.
   - Scegli AWS Data Exchange datashare per creare condivisioni di dati tramite le quali concedere in licenza i tuoi dati. AWS Data Exchange
5. Specificare i valori per Nome unità di condivisione dati, Nome del database e Accessibile al pubblico.

   Quando si modifica il nome del database, è necessario stabilire una nuova connessione al database.

6. Nella sezione Oggetti dell'unità di condivisione dati, scegli Aggiungi. Viene visualizzata la pagina Aggiungi unità di condivisione dati. Per aggiungere oggetti a un'unità di condivisione dati, segui Aggiunta di oggetti di unità di condivisione dati alle unità di condivisione dati.
7. Nella sezione Consumatori di dati, puoi scegliere di pubblicare su un account Redshift o pubblicare su AWS Glue Data Catalog, che avvia il processo di condivisione dei dati tramite Lake Formation. Se pubblichi l'unità di condivisione dati su account Redshift, condividerai i tuoi dati con un altro account Redshift che funge da cluster consumer.

> **Note**
>
> Una volta creata l'unità di condivisione dati, non puoi modificare la configurazione per eseguire la pubblicazione utilizzando l'altra opzione.

8. Quindi scegliere Crea unità di condivisione dati.

Amazon Redshift crea l'unità di condivisione dati. Dopo aver creato l'unità di condivisione dati, sarà possibile creare i database dall'unità.

Aggiunta di oggetti di unità di condivisione dati alle unità di condivisione dati

Aggiungere uno o più oggetti all'unità di condivisione dati. Gli oggetti delle unità di condivisione dati sono di sola lettura per i consumer di dati.

È possibile creare una unità di condivisione dati senza aggiungere oggetti di unità di condivisione dati e aggiungere oggetti in un secondo momento.

Una unità di condivisione dati diventa attiva solo quando si aggiunge almeno un oggetto all'unità.

1. Scegli l'unità di condivisione dati a cui desideri aggiungere oggetti dall'elenco delle unità di condivisione dati.
2. Scegli Aggiungi. Viene visualizzata la pagina Aggiungi oggetti dell'unità di condivisione dati.
3. Prima di aggiungere altri oggetti di unità di condivisione dati, aggiungere almeno uno schema all'unità di condivisione dati. Aggiungere più schemi scegliendo Aggiungi e ripeti.
4. È possibile scegliere di aggiungere tutti gli oggetti esistenti dei tipi di oggetto scelti dallo schema specificato o di aggiungere singoli oggetti specifici dallo schema specificato. Scegli i Tipi di oggetto, come tabelle e viste o funzioni definite dall'utente.
5. È possibile scegliere Aggiungi e ripeti per aggiungere gli schemi e gli oggetti di unità di condivisione dati specificati e continuare ad aggiungere altri oggetti.

Aggiunta di consumer di dati alle unità di condivisione dati

È possibile aggiungere uno o più consumer di dati alle unità di condivisione dati. I consumatori di dati possono essere spazi dei nomi del cluster che identificano in modo univoco i cluster Amazon Redshift o Account AWS.

È necessario scegliere esplicitamente di disabilitare o abilitare l'unità di condivisione dati con cluster con accesso pubblico.

- Scegliere Aggiungi spazi dei nomi del cluster all'unità di condivisione dati. Gli spazi dei nomi sono identificatori univoci globali (GUID) per il cluster Amazon Redshift.
- Scegliere Aggiungi Account AWS all'unità di condivisione dati. L'utente specificato Account AWS deve disporre delle autorizzazioni di accesso al datashare.

Autorizzazione o rimozione dell'autorizzazione dalle unità di condivisione dati

In qualità di amministratore del cluster producer, scegliere quali consumer di dati autorizzare ad accedere alle unità di condivisione dati o da cui rimuovere l'autorizzazione. I consumer di dati autorizzati ricevono notifiche per intraprendere azioni sulle unità di condivisione dati. Se si aggiunge uno spazio dei nomi del cluster come consumer dati, non è necessario eseguire l'autorizzazione.

Prerequisito: per autorizzare o rimuovere l'autorizzazione per l'unità di condivisione dati, è necessario aggiungere almeno un consumer di dati all'unità.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Datashares (Unità di condivisione dati). Viene visualizzata la pagina con l'elenco di unità di condivisione dati.
3. Scegliere Nel mio account.
4. Nella sezione Unità di condivisione dati nel mio account completare una delle operazioni seguenti:
   - Scegli uno o più cluster consumer che desideri autorizzare. Viene visualizzata la pagina Autorizza consumer di dati. Quindi scegliere Authorize (Autorizza).

     Se scegli Pubblica su AWS Glue Data Catalog quando crei l'unità di condivisione dati, puoi fornire l'autorizzazione per l'unità di condivisione dati solo a un account Lake Formation.

     Per quanto riguarda il AWS Data Exchange datashare, puoi autorizzare solo un datashare alla volta.

Quando autorizzi un AWS Data Exchange datashare, condividi il datashare con il AWS Data Exchange servizio e AWS Data Exchange permetti di gestire l'accesso al datashare per tuo conto. AWS Data Exchange consente l'accesso ai consumatori aggiungendo account consumer come consumatori di dati al AWS Data Exchange datashare quando si abbonano ai prodotti. AWS Data Exchange non ha accesso in lettura al datashare.

- Scegli uno o più cluster consumer per i quali desideri rimuovere l'autorizzazione. Quindi scegliere Rimuovi autorizzazione.

Una volta autorizzati i consumer di dati, questi possono accedere agli oggetti dell'unità di condivisione dati e creare un database consumer per eseguire le query sui dati.

Dopo aver rimosso l'autorizzazione, i consumer di dati perderanno immediatamente l'accesso all'unità di condivisione dati.

Gestione delle unità di condivisione dati da altri account in qualità di consumer

Associazione delle unità di condivisione dati

In qualità di amministratore del cluster di consumatori, puoi associare uno o più datashare condivisi da altri account all'intero account o a specifici namespace del cluster nel tuo AWS account.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Datashares (Unità di condivisione dati). Viene visualizzata la pagina con l'elenco di unità di condivisione dati.
3. Scegliere Da altri account.
4. Nella sezione Unità di condivisione dati da altri account, scegliere l'unità di condivisione dati che si desidera associare e scegliere Associa. Quando viene visualizzata la pagina Associa unità di condivisione dati, scegliere uno dei tipi di associazione seguenti:
   - Scegli Intero AWS account per associare tutti i namespace di cluster esistenti e futuri in diverse AWS regioni del tuo AWS account al datashare. Scegliere quindi Associate (Associa).

     Se il datashare è pubblicato su AWS Glue Data Catalog, puoi associarlo solo all'intero account. AWS
   - Scegli AWS Regioni specifiche e namespace del cluster per associare una o più AWS regioni e namespace di cluster specifici al datashare.

a. Scegli Aggiungi regione per aggiungere regioni e namespace di cluster specifici AWS al datashare. Viene visualizzata la pagina Aggiungi regione. AWS

b. Scegliere una Regione AWS .

c. Esegui una di queste operazioni:

- Scegliere Aggiungi tutti gli spazi dei nomi del cluster per aggiungere tutti gli spazi dei nomi del cluster esistenti e futuri in questa regione all'unità di condivisione dati.
- Scegliere Scegli spazi dei nomi specifici del cluster per aggiungere uno o più spazi dei nomi specifici del cluster in questa regione all'unità di condivisione dati.
- Scegli uno o più namespace del cluster e scegli Aggiungi regione. AWS

d. Selezionare Associate (Associa).

Se desideri associare l'unità di condivisione dati a un account Lake Formation, vai alla console Lake Formation per creare un database, quindi definisci le autorizzazioni sul database. Per ulteriori informazioni, consulta Configurazione delle autorizzazioni per le condivisioni di dati Amazon Redshift nella Developer Guide. AWS Lake Formation Dopo aver creato un AWS Glue database o un database federato, puoi utilizzare l'editor di query v2 o qualsiasi client SQL preferito con il tuo cluster di consumatori per interrogare i dati. Per ulteriori informazioni, consulta Utilizzo di unità di condivisione dati gestite da Lake Formation in qualità di consumer.

Dopo aver associato l'unità di condivisione dati, le unità di condivisione dati diventano disponibili.

L'associazione dell'unità di condivisione dati può essere modificata in qualsiasi momento. Quando si modifica l'associazione da AWS regioni e namespace di cluster specifici all'intero AWS account, Amazon Redshift sovrascrive le informazioni specifiche relative alla regione e ai namespace del cluster con le informazioni sull'account. AWS Tutte le AWS regioni e i namespace dei cluster dell'account hanno quindi accesso al datashare. AWS

Quando si modifica l'associazione da spazi di nomi di cluster specifici a tutti gli spazi dei nomi di cluster nella AWS regione specificata, tutti i namespace del cluster in questa regione hanno quindi accesso al datashare.

Rimozione dell'associazione di unità di condivisione dati dai consumer di dati

In qualità di amministratore di cluster consumer, è possibile rimuovere l'associazione di unità di condivisione dati dai consumer di dati.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Datashares (Unità di condivisione dati). Viene visualizzata la pagina con l'elenco di unità di condivisione dati.
3. Scegliere Da altri account.
4. In Unità di condivisione dati da altri account, scegliere l'unità di condivisione dati per rimuovere l'associazione dai consumer di dati.
5. Nella sezione Consumer di dati scegliere uno o più consumer di dati da cui rimuovere l'associazione. Quindi scegliere Rimuovi associazione.
6. Quando viene visualizzata la pagina Rimuovi associazione, scegliere Rimuovi associazione.

Dopo aver rimosso l'autorizzazione, i consumer di dati perderanno immediatamente l'accesso all'unità di condivisione dati. È possibile apportare modifiche all'associazione dei consumer di dati in qualsiasi momento.

Rifiuto delle unità di condivisione dati

In qualità di amministratore del cluster consumer puoi rifiutare qualsiasi unità di condivisione dati il cui stato è disponibile o attiva. Dopo aver rifiutato una unità di condivisione dati, gli utenti del cluster consumer perderanno l'accesso a tale unità di condivisione dati. Amazon Redshift non restituisce l'unità di condivisione dati rifiutata se si richiama l'operazione API `DescribeDataSharesForConsumer`. Se l'amministratore del cluster producer esegue l'operazione API `DescribeDataSharesForProducer` API, vedrà che l'unità di condivisione dati è stata rifiutata. Una volta rifiutato un datashare, l'amministratore del cluster di produttori può autorizzare nuovamente il datashare a un cluster di consumatori e l'amministratore del cluster di consumatori può scegliere di associare il proprio AWS account al datashare o rifiutarlo.

Se il tuo AWS account ha un'associazione a un datashare e un'associazione in sospeso a un datashare gestito da Lake Formation, il rifiuto dell'associazione di datashare gestita da Lake Formation rifiuta anche il datashare originale. Per rifiutare un'associazione specifica, l'amministratore del cluster producer può rimuovere l'autorizzazione da un'unità di condivisione dati specificata. Questa operazione non influisce sulle altre unità di condivisione dati.

Per rifiutare un datashare, utilizza la console, l'operazione API o in. AWS `RejectDataShare` `reject-datashare` AWS CLI

Per rifiutare un datashare utilizzando la console: AWS

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegli Unità di condivisione dati.
3. Scegliere Da altri account.
4. Nella sezione Unità di condivisione dati da altri account, scegliere l'unità di condivisione dati che si desidera rifiutare. Quando viene visualizzata la pagina Rifiuta unità di condivisione dati, scegliere Rifiuta.

L'operazione di rifiuto delle unità di condivisione dati non può essere annullata. Amazon Redshift rimuove le unità di condivisione dati dall'elenco. Per visualizzare nuovamente l'unità di condivisione dati, l'amministratore producer deve autorizzarla nuovamente.

Gestione delle unità di condivisione dati esistenti

Visualizzazione delle unità di condivisione dati

Visualizzare le unità di condivisione dati dalla scheda DATASHARES o CLUSTERS.

- Utilizzare la scheda DATASHARES per elencare le unità di condivisione dati nel proprio account o da altri account.
  - Per visualizzare le unità di condivisione dati create nell'account, selezionare Nel mio account, quindi scegliere l'unità di condivisione dati desiderata.
  - Per visualizzare le unità di condivisione dati condivise da altri account, scegliere Da altri account, quindi scegliere l'unità di condivisione dati desiderata.
- Utilizzare la scheda CLUSTER per elencare le unità di condivisione dati nel cluster o da altri cluster.

  Connettersi a un database. Per ulteriori informazioni, consulta Connessione a un database.

  Quindi scegliere una unità di condivisione dati dalla sezione Unità di condivisione dati da altri cluster o Unità di condivisione dati create nel mio clusterper visualizzarne i dettagli.

Rimozione di oggetti di unità di condivisione dati dalle unità di condivisione dati

È possibile rimuovere uno o più oggetti da un'unità di condivisione dati utilizzando la procedura seguente.

Per rimuovere uno o più oggetti da una unità di condivisione dati

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati.
4. Nella sezione Unità di condivisione dati create nel mio account, scegliere Connetti al database. Per ulteriori informazioni, consulta Connessione a un database.
5. Scegliere l'unità di condivisione dati che si desidera modificare, quindi selezionare Modifica. Viene visualizzata la pagina dei dettagli dell'unità di condivisione dati.
6. Per rimuovere uno o più oggetti di unità di condivisione dati dall'unità di condivisione dati, procedere in uno dei seguenti modi:

   - Per rimuovere gli schemi dall'unità di condivisione dati, scegliere uno o più schemi. Quindi scegliere Rimuovi. Amazon Redshift rimuove gli schemi specificati e tutti gli oggetti degli schemi specificati dall'unità di condivisione dati.
   - Per rimuovere tabelle e viste dall'unità di condivisione dati, scegliere una o più tabelle e viste. Quindi scegliere Rimuovi. In alternativa, scegliere Rimuovi dallo schema per rimuovere tutte le tabelle e tutte le viste negli schemi specificati.
   - Per rimuovere le funzioni definite dall'utente dall'unità di condivisione dati, scegliere una o più funzioni definite dall'utente. Quindi scegliere Rimuovi. In alternativa, scegliere Rimuovi dallo schema per rimuovere tutte le funzioni definite dall'utente negli schemi specificati.

Rimozione dei consumer di dati dalle unità di condivisione dati

È possibile rimuovere uno o più consumer di dati da una unità di condivisione dati. I consumatori di dati possono essere namespace di cluster che identificano in modo univoco cluster o account Amazon Redshift. AWS

Scegli uno o più consumatori di dati dagli ID o dall'account dello spazio dei nomi del cluster, quindi scegli Rimuovi. AWS

Amazon Redshift rimuove i consumer di dati specificati dall'unità di condivisione dati. L'accesso all'unità di condivisione dati si perde immediatamente.

Modifica delle unità di condivisione dati create nell'account

Modifica le unità di condivisione dati create nell'account mediante la console. Connettersi prima a un database per visualizzare l'elenco delle unità di condivisione dati create nell'account.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati.
4. Nella sezione Unità di condivisione dati create nel mio account, scegliere Connetti al database. Per ulteriori informazioni, consulta Connessione a un database.
5. Scegliere l'unità di condivisione dati che si desidera modificare, quindi selezionare Modifica. Viene visualizzata la pagina dei dettagli dell'unità di condivisione dati.
6. Apportare eventuali modifiche nella finestra di dialogoOggetti unità di condivisione dati o nella sezione Consumer di dati.

> **Note**
>
> Se hai scelto di pubblicare il tuo datashare su AWS Glue Data Catalog, non puoi modificare la configurazione per pubblicare il datashare su altri account Amazon Redshift.

7. Seleziona Salvataggio delle modifiche.

Amazon Redshift aggiorna l'unità di condivisione dati con le modifiche.

Eliminazione delle unità di condivisione dati create nell'account

Eliminare le unità di condivisione dati create nell'account mediante la console. Connettersi prima a un database per visualizzare l'elenco delle unità di condivisione dati create nell'account.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati. Viene visualizzato un elenco di unità di condivisione dati.

4. Nella sezione Unità di condivisione dati create nel mio account, scegliere Connetti al database. Per ulteriori informazioni, consulta Connessione a un database.
5. Scegliere una o più unità di condivisione dati da eliminare, quindi scegliere Elimina. Viene visualizzata la pagina Elimina unità di condivisione dati.

   L'eliminazione di un'unità di condivisione dati condivisa con Lake Formation non rimuove automaticamente le autorizzazioni associate in Lake Formation. Per rimuoverle, vai alla console di Lake Formation.
6. Digitare Elimina per confermare l'eliminazione delle unità di condivisione dati specificate.
7. Scegli Elimina.

Una volta eliminate le unità di condivisione dati, i consumer delle unità di condivisione dati perderanno l'accesso alle unità.

Esecuzione di query sulle unità di condivisione dati

Creazione di database da unità di condivisione dati

Per iniziare a interrogare i dati nell'unità di condivisione dati, creare database da una unità. È possibile creare un solo database da una unità di condivisione dati specificata.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati. Viene visualizzato un elenco di unità di condivisione dati.
4. Nella sezione Unità di condivisione dati da altri cluster, scegliere Connetti al database. Per ulteriori informazioni, consulta Connessione a un database.
5. Scegliere una unità di condivisione dati da cui si desidera creare i database, quindi selezionare Crea database da unità di condivisione dati. Viene visualizzata la pagina Crea database da unità di condivisione dati.
6. In Nome del database, specificare un nome per il database. Il nome del database deve contenere un numero di caratteri alfanumerici (solo minuscoli) compreso tra 1 e 64 caratteri e non può essere una parola riservata.
7. Scegli Crea.

Dopo aver creato il database, sarà possibile eseguire una query sui dati nel database.

Gestione delle condivisioni di dati AWS Data Exchange

Creazione di set di dati su AWS Data Exchange

Crea set di dati su AWS Data Exchange.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati.
4. Nella sezione Datashare create nel mio account, scegli un datashare. AWS Data Exchange
5. Scegli Crea set di dati su. AWS Data Exchange Per ulteriori informazioni, consultare Pubblicazione di un nuovo prodotto.

Modifica delle condivisioni AWS Data Exchange di dati

Modifica le AWS Data Exchange condivisioni di dati utilizzando la console. Connettersi prima a un database per visualizzare l'elenco delle unità di condivisione dati create nell'account.

Per quanto riguarda le AWS Data Exchange condivisioni di dati, non è possibile apportare modifiche ai consumatori di dati.

Per modificare l'impostazione accessibile pubblicamente per le AWS Data Exchange condivisioni di dati, usa l'editor di Query v2. Amazon Redshift genera un valore casuale una tantum per impostare la variabile di sessione in modo da consentire di disattivare questa impostazione. Per ulteriori informazioni, consulta Note per l'utilizzo di ALTER DATASHARE.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Dal menu navigator, selezionare Editor, quindi Editor di query v2.
4. Se è la prima volta che usi l'editor di query v2, configura il tuo Account AWS. Per impostazione predefinita, viene utilizzata una chiave AWS proprietaria per crittografare le risorse. Per ulteriori

informazioni sulla configurazione del tuo Account AWS, consulta Configuring your Account AWS nella Amazon Redshift Management Guide.

5. Per connetterti al cluster in cui si trova il tuo AWS Data Exchange datashare, scegli Database e il nome del cluster nel pannello di visualizzazione ad albero. Se richiesto, immettere i parametri di connessione.
6. Utilizza la seguente istruzione SQL. L'esempio seguente modifica l'impostazione accessibile al pubblico dell'unità di condivisione dati di salesshare.

```
ALTER DATASHARE salesshare SET PUBLICACCESSIBLE FALSE;
```

7. Per eseguire l'istruzione SQL copiata, scegliere Query e incolla l'istruzione SQL copiata nell'area della query. Quindi scegli Esegui.

   Viene visualizzato l'errore seguente:

```
ALTER DATASHARE salesshare SET PUBLICACCESSIBLE FALSE;
ERROR:  Alter of ADX-managed datashare salesshare requires session variable
 datashare_break_glass_session_var to be set to value 'c670ba4db22f4b'
```

   Il valore 'c670ba4db22f4b' è un valore occasionale casuale generato da Amazon Redshift quando si verifica un'operazione non consigliata.
8. Copiare e incollare la seguente istruzione esempio nell'editor di query. Quindi eseguire il comando. Il `SET datashare_break_glass_session_var` comando applica un'autorizzazione per consentire un'operazione non consigliata per un datashare. AWS Data Exchange

```
SET datashare_break_glass_session_var to 'c670ba4db22f4b';
```

9. Esegui di nuovo l'istruzione ALTER DATASHARE.

```
ALTER DATASHARE salesshare;
```

Amazon Redshift aggiorna l'unità di condivisione dati con le modifiche.

Eliminazione delle condivisioni di AWS Data Exchange dati create nell'account

Elimina le AWS Data Exchange condivisioni di dati create nel tuo account utilizzando la console. Connettersi prima a un database per visualizzare l'elenco delle unità di condivisione dati create nell'account.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Dal menu navigator, selezionare Editor, quindi Editor di query v2.
4. Se è la prima volta che usi l'editor di query v2, configura il tuo Account AWS. Per impostazione predefinita, viene utilizzata una chiave AWS proprietaria per crittografare le risorse. Per ulteriori informazioni sulla configurazione del tuo Account AWS, consulta Configuring your Account AWS nella Amazon Redshift Management Guide.
5. Per connetterti al cluster in cui si trova il tuo AWS Data Exchange datashare, scegli Database e il nome del cluster nel pannello di visualizzazione ad albero. Se richiesto, immettere i parametri di connessione.
6. Utilizza la seguente istruzione SQL. L'esempio seguente elimina l'unità di condivisione dati denominata salesshare.

```
DROP DATASHARE salesshare
```

7. Per eseguire l'istruzione SQL copiata, scegliere Query e incolla l'istruzione SQL copiata nell'area della query. Quindi scegli Esegui.

   Viene visualizzato l'errore seguente:

```
ERROR:  Drop of ADX-managed datashare salesshare requires session variable
 datashare_break_glass_session_var to be set to value '620c871f890c49'
```

   Il valore '620c871f890c49' è un valore occasionale casuale generato da Amazon Redshift quando si verifica un'operazione non consigliata.
8. Copiare e incollare la seguente istruzione esempio nell'editor di query. Quindi eseguire il comando. Il `SET datashare_break_glass_session_var` comando applica un'autorizzazione per consentire un'operazione non consigliata per un datashare. AWS Data Exchange

```
SET datashare_break_glass_session_var to '620c871f890c49';
```

9. Esegui di nuovo l'istruzione DROP DATASHARE.

```
DROP DATASHARE salesshare;
```

Una volta eliminate le unità di condivisione dati, i consumatori delle unità di condivisione dati perderanno l'accesso alle unità.

L'eliminazione di un AWS Data Exchange datashare condiviso può violare i termini del prodotto di dati in. AWS Data Exchange

Gestione della condivisione dati con AWS CloudFormation

Puoi automatizzare la configurazione della condivisione dei dati utilizzando uno AWS CloudFormation stack che fornisce risorse. AWS Lo CloudFormation stack configura la condivisione dei dati tra due cluster Amazon Redshift nello stesso account. AWS In questo modo, puoi avviare la condivisione dati senza eseguire istruzioni SQl per il provisioning delle risorse.

Lo stack crea una unità di condivisione dati sul cluster designato. L'unità di condivisione dati include una tabella e dati di esempio di sola lettura. Questi dati possono essere letti dall'altro cluster Amazon Redshift.

Se desideri iniziare a condividere i dati in un AWS account eseguendo istruzioni SQL per configurare un datashare e concedere le autorizzazioni, senza utilizzarlo, consulta. CloudFormation Condivisione dell'accesso in lettura ai dati all'interno di un Account AWS

Prima di eseguire lo CloudFormation stack di condivisione dei dati, devi accedere con un utente autorizzato a creare un ruolo IAM e una funzione Lambda. Hai anche bisogno di due cluster Amazon Redshift nello stesso account. Tu ne usi uno, il produttore, per condividere i dati di esempio e l'altro consumatore, per leggerlo. Il requisito principale per questi cluster è che ognuno utilizza nodi RA3. Per i requisiti aggiuntivi, consultare Considerazioni sull'utilizzo della condivisione dei dati in Amazon Redshift.

Per ulteriori informazioni su come iniziare a configurare un cluster Amazon Redshift, consulta Amazon Redshift provisioned clusters. Per ulteriori informazioni sull'automazione della configurazione con CloudFormation, consulta What is? AWS CloudFormation

> **Important**
>
> Prima di lanciare lo CloudFormation stack, assicurati di avere due cluster Amazon Redshift nello stesso account e che i cluster utilizzino nodi RA3. Assicurati che ogni cluster abbia un database e un utente con privilegi avanzati. Per ulteriori informazioni, consulta CREATE DATABASE e superuser.

Per avviare CloudFormation lo stack per la condivisione dei dati di Amazon Redshift:

1. Fai clic su Avvia lo stack CFN, che ti porterà al CloudFormation servizio in. AWS Management Console

   Se ti viene richiesto, effettua l'accesso.

   Viene avviato il processo di creazione dello stack, facendo riferimento a un file CloudFormation modello archiviato in Amazon S3. Un CloudFormation modello è un file di testo in formato JSON che dichiara AWS le risorse che compongono uno stack. Per ulteriori informazioni sui CloudFormation modelli, consulta Impara le nozioni di base sui modelli.
2. Scegliere Successivo per inserire i dettagli della pila.
3. Sotto Parametri, per ciascun cluster, immettere quanto segue:

   - Il nome del cluster Amazon Redshift, ad esempio **`ra3-consumer-cluster`**
   - Il nome del database, ad esempio **`dev`**
   - Il nome dell'utente del database, ad esempio **`consumeruser`**

   Si consiglia di utilizzare cluster di test, perché la pila crea diversi oggetti di database.

   Seleziona Successivo.
4. Vengono visualizzate le opzioni della pila.

   Scegliere Successivo per accettare le impostazioni predefinite.
5. In Capacità, scegli Riconosco che AWS CloudFormation potrebbe creare risorse IAM.
6. Seleziona Crea stack.

CloudFormation impiega circa 10 minuti per creare lo stack Amazon Redshift utilizzando il modello, creando un datashare chiamato. `myproducer_share` La pila crea l'unità nel database specificato nei dettagli della pila. Solo gli oggetti di quel database possono essere condivisi.

Se si verifica un errore durante la creazione della pila, procedere come segue:

- Assicurarsi di aver immesso il nome del cluster, il nome del database e il nome utente del database corretti per ciascun cluster Redshift.
- Assicurarsi che il cluster disponga di nodi RA3.

- Assicurati di essere collegato come un utente che abbia l'autorizzazione a creare un ruolo IAM e una funzione Lambda. Per ulteriori informazioni sulla creazione dei ruoli IAM, consulta Creazione di ruoli IAM. Per ulteriori informazioni sulle policy per la creazione di funzioni Λ, consulta Function development.

Query del datashare creato

Per utilizzare la procedura seguente, assicurarsi di disporre delle autorizzazioni necessarie per eseguire query su ciascun cluster descritto.

Per eseguire query sull'unità unità di condivisione dati:

1. Connettiti al cluster di produttori sul database inserito al momento della creazione CloudFormation dello stack, utilizzando uno strumento client come Amazon Redshift query editor v2.
2. Esegui query sulle unità di condivisione dati.

```
SHOW DATASHARES;

+------------------+-------------+-----------------+------------------
+------------+------------+--------------------+-----------+-----------------
+--------------------------------------+
|    share_name    | share_owner | source_database | consumer_database | share_type
 | createdate | is_publicaccessible | share_acl | producer_account |
 producer_namespace          |
+------------------+-------------+-----------------+------------------
+------------+------------+--------------------+-----------+-----------------
+--------------------------------------+
| myproducer_share | 100         | sample_data_dev | myconsumer_db     | INBOUND
  | NULL       | true               | NULL      |  producer-acct   |        your-
producer-namespace       |
+------------------+-------------+-----------------+------------------
+------------+------------+--------------------+-----------+-----------------
+--------------------------------------+
```

Il comando precedente restituisce il nome dell'unità creata dalla pila, denominata `myproducer_share`. Restituisce inoltre il nome del database associato all'unità di condivisione dati, .,`myconsumer_db`.

Copiare l'identificatore dello spazio dei nomi del produttore da utilizzare in un passaggio successivo.

3. Descrivi gli oggetti nell'unità unità di condivisione dati.

```
DESC DATASHARE myproducer_share;

+------------------+--------------------------------------+------------
+------------------+-------------+-------------------------------------
+-------------+
| producer_account |          producer_namespace          | share_type |
 share_name    | object_type |             object_name             | include_new |
+------------------+--------------------------------------+------------
+------------------+-------------+-------------------------------------
+-------------+
|   producer-acct  |        your-producer-namespace       | OUTBOUND   |
 myproducer_share | schema      | myproducer_schema                   | true
 |
|   producer-acct  |        your-producer-namespace       | OUTBOUND   |
 myproducer_share | table       | myproducer_schema.tickit_sales      | NULL
 |
|   producer-acct  |        your-producer-namespace       | OUTBOUND   |
 myproducer_share | view        | myproducer_schema.ticket_sales_view | NULL
 |
+------------------+--------------------------------------+------------
+------------------+-------------+-------------------------------------
+-------------+
```

Quando descrivi l'unità di condivisione dati, vengono restituite le proprietà per tabelle e viste. La pila aggiunge tabelle e viste con dati di esempio al database del produttore, ad esempio `tickit_sales` e `tickit_sales_view`. Per ulteriori informazioni sul database di esempio TICKIT, consultaDatabase di esempio.

Non è necessario delegare le autorizzazioni nell'unità per eseguire query. La pila concede le autorizzazioni necessarie.

4. Connettitti al cluster consumer utilizzando lo strumento client. Descrivi l'unità, specificando lo spazio dei nomi del produttore.

```
DESC DATASHARE myproducer_share OF NAMESPACE '<namespace id>'; --specify the unique
 identifier for the producer namespace
```

```
+-----------------+--------------------------------------+-----------
+-----------------+-------------+-----------------------------------
+-------------+
| producer_account |          producer_namespace          | share_type |
 share_name    | object_type |             object_name             | include_new |
+-----------------+--------------------------------------+-----------
+-----------------+-------------+-----------------------------------
+-------------+
|   producer-acct  |        your-producer-namespace       | INBOUND    |
 myproducer_share | schema      | myproducer_schema                   | NULL
 |
|   producer-acct  |        your-producer-namespace       | INBOUND    |
 myproducer_share | table       | myproducer_schema.tickit_sales      | NULL
 |
|   producer-acct  |        your-producer-namespace       | INBOUND    |
 myproducer_share | view        | myproducer_schema.ticket_sales_view | NULL
 |
+-----------------+--------------------------------------+-----------
+-----------------+-------------+-----------------------------------
+-------------+
```

5. È possibile eseguire query sulle tabelle nell'unità unità di condivisione dati specificando il database e lo schema dell'unità. Per ulteriori informazioni, consulta Esempi di utilizzo di una query tra database. Le seguenti query restituiscono i dati relativi alle vendite e ai venditori dalla tabella SALES del database di esempio TICKIT. Per ulteriori informazioni, consulta Tabella SALES.

```
SELECT * FROM myconsumer_db.myproducer_schema.tickit_sales_view;

+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
| salesid | listid | sellerid | buyerid | eventid | dateid | qtysold | pricepaid |
 commission |      saletime       |
+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
|       1 |      1 |    36861 |   21191 |    7872 |   1875 |       4 |       728 |
     109.2 | 2008-02-18 02:36:48 |
|       2 |      4 |     8117 |   11498 |    4337 |   1983 |       2 |        76 |
      11.4 | 2008-06-06 05:00:16 |
|       3 |      5 |     1616 |   17433 |    8647 |   1983 |       2 |       350 |
      52.5 | 2008-06-06 08:26:17 |
|       4 |      5 |     1616 |   19715 |    8647 |   1986 |       1 |       175 |
     26.25 | 2008-06-09 08:38:52 |
```

```
|       5 |      6 |    47402 |   14115 |    8240 |   2069 |       2 |       154 |
      23.1 | 2008-08-31 09:17:02 |
+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
```

> **Note**
>
> La query viene eseguita contro la vista nello schema condiviso. Non è possibile connettersi direttamente ai database consumer creati dalle unità di condivisione dati. Sono di sola lettura.

6. Per eseguire una query che include aggregazioni, utilizza l'esempio seguente.

```
SELECT * FROM myconsumer_db.myproducer_schema.tickit_sales ORDER BY 1,2 LIMIT 5;

+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
| salesid | listid | sellerid | buyerid | eventid | dateid | qtysold | pricepaid |
 commission |      saletime       |
+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
|       1 |      1 |    36861 |   21191 |    7872 |   1875 |       4 |       728 |
     109.2 | 2008-02-18 02:36:48 |
|       2 |      4 |     8117 |   11498 |    4337 |   1983 |       2 |        76 |
      11.4 | 2008-06-06 05:00:16 |
|       3 |      5 |     1616 |   17433 |    8647 |   1983 |       2 |       350 |
      52.5 | 2008-06-06 08:26:17 |
|       4 |      5 |     1616 |   19715 |    8647 |   1986 |       1 |       175 |
     26.25 | 2008-06-09 08:38:52 |
|       5 |      6 |    47402 |   14115 |    8240 |   2069 |       2 |       154 |
      23.1 | 2008-08-31 09:17:02 |
+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
```

La query restituisce i dati delle vendite e del venditore dai dati del database di esempio TICKIT.

Per ulteriori esempi di query di unità di condivisione dati, consultare Condivisione dell'accesso in lettura ai dati all'interno di un Account AWS.

Gestione della condivisione dei dati con operazioni di scrittura tramite la console (anteprima)

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Per ulteriori informazioni sulla configurazione della traccia PREVIEW_2023, consulta una delle seguenti pagine:

- Per l'anteprima di Amazon Redshift serverless: Creazione di un gruppo di lavoro di anteprima
- Per l'anteprima dei cluster con provisioning di Amazon Redshift: Creazione di un cluster di anteprima

Per ulteriori informazioni su come iniziare a condividere i dati, vai a Condivisione, scrittura, accesso ai dati (anteprima).

Utilizzare la console Amazon Redshift per gestire i dati creati nell'account o condivisi da altri account.

Connessione a un database (anteprima)

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Per ulteriori informazioni su come iniziare a condividere i dati, vai a Condivisione dell'accesso in scrittura ai dati (anteprima).

Connettersi a un database per visualizzare i database e gli oggetti all'interno dei database in questo cluster o per visualizzare le unità di condivisione dati.

Le credenziali utente utilizzate per connettersi a un database specificato devono disporre delle autorizzazioni necessarie per visualizzare tutte le unità di condivisione dati.

Se non esiste una connessione locale, procedere in uno dei seguenti modi:

- Nella pagina dei dettagli del cluster, dalla tabella Databases (Database), nella sezione Databases (Database) o Datashare objects (Oggetti dell'unità di condivisione dati), scegli Connect to database (Connetti al database) per visualizzare gli oggetti di database nel cluster.
- Nella pagina dei dettagli del cluster, dalla scheda Unità di condivisione dati effettuare una delle seguenti operazioni:
  - Nella sezione Unità di condivisione dati da altri cluster, scegliere Connetti al database per visualizzare le unità di condivisione dati da altri cluster.
  - Nella sezione Unità di condivisione dati da altri cluster, scegliere Connetti al database per visualizzare le unità di condivisione dati da altri cluster.
- Nella finestra Connetti al database, completare una delle seguenti operazioni:
  - Se si sceglie Crea una nuova connessione, scegliere AWS Secrets Manager per utilizzare un segreto memorizzato per autenticare l'accesso per la connessione.

    In alternativa, scegliere Credenziali temporanee per utilizzare le credenziali del database. Specificare i valori per Nome database e Utente database.

    Scegli Connetti.
  - Scegliere Usa una connessione recente per connettersi a un altro database per cui si dispone delle autorizzazioni necessarie.

    Amazon Redshift effettua automaticamente la connessione.

Dopo aver stabilito la connessione al database, è possibile iniziare a creare unità di condivisione dati, eseguire query sulle unità di condivisione dati o creare database dalle unità di condivisione dati.

Creazione di unità di condivisione dati e aggiunta di oggetti (anteprima)

Creazione di unità di condivisione dati

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Per ulteriori informazioni su come iniziare a condividere i dati, vai a Condivisione dell'accesso in scrittura ai dati (anteprima).

In qualità di amministratore del cluster producer, è possibile creare unità di condivisione dati dalle schede Database o Unità di condivisione dati nella pagina dei dettagli del cluster.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Nella pagina dei dettagli del cluster, effettuare una delle seguenti operazioni:
   - Dalla scheda Database, nella sezione Database scegliere un database. Viene visualizzata la pagina dei dettagli del database.

     Quindi scegliere Crea unità di condivisione dati. È possibile creare una unità di condivisione dati solo da un database locale. Se è la prima volta che si esegue la connessione al database, viene visualizzata la pagina Connetti al database. Seguire la procedura riportata in Connessione a un database (anteprima) per connettersi a un database. Se esiste una connessione recente, viene visualizzata la pagina Crea unità di condivisione dati.
   - Dalla scheda Unità di condivisione dati, nella scheda Unità di condivisione dati, connettersi a un database se non si dispone di una connessione al database.

     Nella sezione Unità di condivisione dati create nel mio cluster, scegliere Crea unità di condivisione dati. Viene visualizzata la pagina Crea unità di condivisione dati.

4. In questa pagina puoi aggiungere oggetti di database di vario tipo. Seleziona il pulsante Aggiungi per aggiungere gli oggetti. Viene visualizzata una finestra di dialogo. Esegui queste fasi:

   1. Scegli uno o più schemi. In questo modo gli oggetti degli schemi sono disponibili per essere aggiunti.
   2. Seleziona Tipi di oggetto negli schemi.

      Qui puoi scegliere un paio di opzioni per aggiungere gli oggetti:

      - Aggiungi oggetti specifici di schemi: se scegli questa opzione, vengono elencati i singoli oggetti per nome. Puoi selezionare gli oggetti e aggiungerli all'unità di condivisione dati. Ad esempio, puoi aggiungere tabelle e stored procedure specifiche, se lo desideri. Quindi, le tabelle e le stored procedure dello schema selezionato vengono incluse nell'unità di condivisione dati. L'impostazione delle autorizzazioni viene spiegata in passaggi successivi. Continua a selezionare gli oggetti da aggiungere con Viste e altri tipi.
      - Aggiungi allo schema tutti gli oggetti esistenti dai tipi di oggetto selezionati: con questa opzione vengono aggiunti tutti gli oggetti.
   3. Puoi anche scegliere di aggiungere oggetti futuri. Quando scegli di includere gli oggetti dell'unità di condivisione dati aggiunti allo schema significa che gli oggetti aggiunti allo schema vengono aggiunti automaticamente all'unità di condivisione dati.
   4. Scegli Aggiungi per completare la sezione e aggiungere gli oggetti. Sono elencati in Oggetti della condivisione di dati.
   5. Dopo averli aggiunti, puoi selezionare singoli oggetti e modificarne le autorizzazioni. Se selezioni uno schema, viene visualizzata una finestra di dialogo che ti chiede se desideri aggiungere le autorizzazioni con ambito. In tal modo ogni oggetto esistente o aggiunto allo schema dispone di un set di autorizzazioni preselezionato e appropriato per il tipo di oggetto. Ad esempio, l'amministratore può impostare che tutte le tabelle aggiunte abbiano le autorizzazioni SELECT e UPDATE.
   6. Dopo aver configurato le autorizzazioni dello schema, puoi esaminare altri tipi di oggetti e selezionare le relative autorizzazioni. Ad esempio, puoi aggiungere le autorizzazioni UPDATE a una tabella specifica.
   7. Nella sezione Consumatori di dati, puoi aggiungere namespace o aggiungere AWS account come consumatori del datashare.
   8. Per salvare le modifiche seleziona Crea unità di condivisione dati.

Una volta creata, l'unità di condivisione dati viene visualizzata nell'elenco in Unità di condivisione dati create nel mio spazio dei nomi. Se scegli un'unità di condivisione dati dall'elenco, puoi visualizzarne i consumer, gli oggetti e altre proprietà.

Aggiunta di consumer di dati alle unità di condivisione dati

È possibile aggiungere uno o più consumer di dati alle unità di condivisione dati. I consumatori di dati possono essere spazi dei nomi del cluster che identificano in modo univoco i cluster Amazon Redshift o Account AWS.

È necessario scegliere esplicitamente di disabilitare o abilitare l'unità di condivisione dati con cluster con accesso pubblico.

- Scegliere Aggiungi spazi dei nomi del cluster all'unità di condivisione dati. Gli spazi dei nomi sono identificatori univoci globali (GUID) per il cluster Amazon Redshift.
- Scegliere Aggiungi Account AWS all'unità di condivisione dati. L'utente specificato Account AWS deve disporre delle autorizzazioni di accesso al datashare.

Autorizzazione o rimozione dell'autorizzazione dalle unità di condivisione dati (anteprima)

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Per ulteriori informazioni su come iniziare a condividere i dati, vai a Condivisione, scrittura, accesso ai dati (anteprima).

In qualità di amministratore del cluster producer, scegliere quali consumer di dati autorizzare ad accedere alle unità di condivisione dati o da cui rimuovere l'autorizzazione. I consumer di dati autorizzati ricevono notifiche per intraprendere azioni sulle unità di condivisione dati. Se si aggiunge uno spazio dei nomi del cluster come consumer dati, non è necessario eseguire l'autorizzazione.

Prerequisito: per autorizzare o rimuovere l'autorizzazione per l'unità di condivisione dati, è necessario aggiungere almeno un consumer di dati all'unità.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Datashares (Unità di condivisione dati). Viene visualizzato l'elenco Consumer delle unità di condivisione dati. Scegli uno o più cluster consumer che desideri autorizzare. Quindi scegliere Authorize (Autorizza).
3. Viene visualizzata la finestra di dialogo Autorizza account. Puoi scegliere tra un paio di tipi di autorizzazione.

   - Sola lettura su [nome del cluster o nome del gruppo di lavoro]: significa che le autorizzazioni di scrittura non sono disponibili per il consumer, anche se il creatore dell'unità di condivisione dati le ha assegnate.
   - Lettura e scrittura su [nome del cluster o nome del gruppo di lavoro]: significa che tutte le autorizzazioni assegnate dal creatore, incluse le autorizzazioni di scrittura, sono disponibili per il consumer.
4. Selezionare Salva.

Puoi anche effettuare l'autorizzazione AWS Data Exchange come consumatore.

1. Se scegli Pubblica su AWS Glue Data Catalog quando crei l'unità di condivisione dati, puoi fornire l'autorizzazione per l'unità di condivisione dati solo a un account Lake Formation.

   Per il AWS Data Exchange datashare, puoi autorizzare solo un datashare alla volta.

   Quando autorizzi un AWS Data Exchange datashare, condividi il datashare con il AWS Data Exchange servizio e AWS Data Exchange permetti di gestire l'accesso al datashare per tuo conto. AWS Data Exchange consente l'accesso ai consumatori aggiungendo account consumer come consumatori di dati al AWS Data Exchange datashare quando si abbonano ai prodotti. AWS Data Exchange non ha accesso in lettura al datashare.
2. Selezionare Salva.

Una volta autorizzati i consumer di dati, questi possono accedere agli oggetti dell'unità di condivisione dati e creare un database consumer per eseguire le query sui dati.

Rimozione dell'autorizzazione:

Scegli uno o più cluster consumer per i quali desideri rimuovere l'autorizzazione. Quindi scegliere Rimuovi autorizzazione.

Dopo aver rimosso l'autorizzazione, i consumer di dati perderanno immediatamente l'accesso all'unità di condivisione dati.

Associazione o rifiuto delle unità di condivisione dati come consumer (anteprima)

Associazione delle unità di condivisione dati

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Per ulteriori informazioni su come iniziare a condividere i dati, vai a Condivisione, scrittura, accesso ai dati (anteprima).

In qualità di amministratore di un cluster di consumatori, puoi associare una o più condivisioni di dati condivise da altri account all'intero AWS account o a specifici namespace del cluster nel tuo account.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Datashares (Unità di condivisione dati). Viene visualizzata la pagina con l'elenco di unità di condivisione dati. Scegliere Da altri account.
3. Nella sezione Unità di condivisione dati da altri account, scegliere l'unità di condivisione dati che si desidera associare e scegliere Associa. Quando viene visualizzata la pagina Associa unità di condivisione dati scegli uno dei tipi di associazione seguenti:

   - Scegli Intero AWS account per associare tutti i namespace di cluster esistenti e futuri in diverse AWS regioni del tuo AWS account al datashare.

     Se il datashare è pubblicato su AWS Glue Data Catalog, puoi associarlo solo all'intero account. AWS
4. Qui puoi scegliere Autorizzazioni consentite. Le scelte disponibili sono le seguenti:

- Sola lettura: se scegli questa opzione, le autorizzazioni di scrittura come UPDATE o INSERT non sono disponibili per il consumer, anche sono state assegnate e autorizzate dal producer.
- Lettura e scrittura: gli utenti dell'unità di condivisione dati del consumer dispongono di tutte le autorizzazioni di lettura e di scrittura, assegnate e autorizzate dal produttore.

5. Oppure scegli AWS Regioni specifiche e namespace del cluster per associare una o più AWS regioni e namespace di cluster specifici al datashare. Scegli Aggiungi regione per aggiungere regioni e namespace di cluster specifici AWS al datashare. Viene visualizzata la pagina Aggiungi regione. AWS
6. Scegliere una Regione AWS .
7. Esegui una di queste operazioni:
   - Scegliere Aggiungi tutti gli spazi dei nomi del cluster per aggiungere tutti gli spazi dei nomi del cluster esistenti e futuri in questa regione all'unità di condivisione dati.
   - Scegliere Scegli spazi dei nomi specifici del cluster per aggiungere uno o più spazi dei nomi specifici del cluster in questa regione all'unità di condivisione dati.
   - Scegli uno o più namespace del cluster e scegli Aggiungi regione. AWS
8. Selezionare Associate (Associa).

Il producer può tornare indietro e modificare le impostazioni di un'autorizzazione, il che può influire sulle impostazioni di associazione dei consumer.

Se desideri associare l'unità di condivisione dati a un account Lake Formation, vai alla console Lake Formation per creare un database, quindi definisci le autorizzazioni sul database. Per ulteriori informazioni, consulta Configurazione delle autorizzazioni per le condivisioni di dati Amazon Redshift nella Developer Guide. AWS Lake Formation Dopo aver creato un AWS Glue database o un database federato, puoi utilizzare l'editor di query v2 o qualsiasi client SQL preferito con il tuo cluster di consumatori per interrogare i dati.

Dopo aver associato l'unità di condivisione dati, le unità di condivisione dati diventano disponibili.

L'associazione dell'unità di condivisione dati può essere modificata in qualsiasi momento. Quando si modifica l'associazione da AWS regioni e namespace di cluster specifici all'intero AWS account, Amazon Redshift sovrascrive le informazioni specifiche relative alla regione e ai namespace del cluster con le informazioni sull'account. AWS Tutte le AWS regioni e i namespace dei cluster dell'account hanno quindi accesso al datashare. AWS

Quando si modifica l'associazione da spazi di nomi di cluster specifici a tutti gli spazi dei nomi di cluster nella AWS regione specificata, tutti i namespace del cluster in questa regione hanno quindi accesso al datashare.

Rimozione dell'associazione di unità di condivisione dati dai consumer di dati

In qualità di amministratore di cluster consumer, è possibile rimuovere l'associazione di unità di condivisione dati dai consumer di dati.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Datashares (Unità di condivisione dati). Viene visualizzata la pagina con l'elenco di unità di condivisione dati.
3. Scegliere Da altri account.
4. In Unità di condivisione dati da altri account, scegliere l'unità di condivisione dati per rimuovere l'associazione dai consumer di dati.
5. Nella sezione Consumer di dati scegliere uno o più consumer di dati da cui rimuovere l'associazione. Quindi scegliere Rimuovi associazione.
6. Quando viene visualizzata la pagina Rimuovi associazione, scegliere Rimuovi associazione.

Dopo aver rimosso l'autorizzazione, i consumer di dati perderanno immediatamente l'accesso all'unità di condivisione dati. È possibile apportare modifiche all'associazione dei consumer di dati in qualsiasi momento.

Rifiuto delle unità di condivisione dati

In qualità di amministratore del cluster consumer puoi rifiutare qualsiasi unità di condivisione dati il cui stato è disponibile o attiva. Dopo aver rifiutato una unità di condivisione dati, gli utenti del cluster consumer perderanno l'accesso a tale unità di condivisione dati. Amazon Redshift non restituisce l'unità di condivisione dati rifiutata se si richiama l'operazione API `DescribeDataSharesForConsumer`. Se l'amministratore del cluster producer esegue l'operazione API `DescribeDataSharesForProducer` API, vedrà che l'unità di condivisione dati è stata rifiutata. Una volta rifiutato un datashare, l'amministratore del cluster di produttori può autorizzare nuovamente il datashare a un cluster di consumatori e l'amministratore del cluster di consumatori può scegliere di associare il proprio AWS account al datashare o rifiutarlo.

Se il tuo AWS account ha un'associazione a un datashare e un'associazione in sospeso a un datashare gestito da Lake Formation, il rifiuto dell'associazione di datashare gestita da Lake

Formation rifiuta anche il datashare originale. Per rifiutare un'associazione specifica, l'amministratore del cluster producer può rimuovere l'autorizzazione da un'unità di condivisione dati specificata. Questa operazione non influisce sulle altre unità di condivisione dati.

Per rifiutare un datashare, utilizza la console, l'operazione API o in. AWS `RejectDataShare` `reject-datashare` AWS CLI

Per rifiutare un datashare utilizzando la console: AWS

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegli Unità di condivisione dati.
3. Scegliere Da altri account.
4. Nella sezione Unità di condivisione dati da altri account, scegliere l'unità di condivisione dati che si desidera rifiutare. Quando viene visualizzata la pagina Rifiuta unità di condivisione dati, scegliere Rifiuta.

L'operazione di rifiuto delle unità di condivisione dati non può essere annullata. Amazon Redshift rimuove le unità di condivisione dati dall'elenco. Per visualizzare nuovamente l'unità di condivisione dati, l'amministratore producer deve autorizzarla nuovamente.

Gestione delle unità di condivisione dati esistenti (anteprima)

> Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Per ulteriori informazioni su come iniziare a condividere i dati, vai a Condivisione, scrittura, accesso ai dati (anteprima).

Visualizzazione delle unità di condivisione dati

Visualizzare le unità di condivisione dati dalla scheda DATASHARES o CLUSTERS.

- Utilizzare la scheda DATASHARES per elencare le unità di condivisione dati nel proprio account o da altri account.
  - Per visualizzare le unità di condivisione dati create nell'account, selezionare Nel mio account, quindi scegliere l'unità di condivisione dati desiderata.
  - Per visualizzare le unità di condivisione dati condivise da altri account, scegliere Da altri account, quindi scegliere l'unità di condivisione dati desiderata.
- Utilizzare la scheda CLUSTER per elencare le unità di condivisione dati nel cluster o da altri cluster.

  Connettersi a un database. Per ulteriori informazioni, consulta Connessione a un database (anteprima).

  Quindi scegliere una unità di condivisione dati dalla sezione Unità di condivisione dati da altri cluster o Unità di condivisione dati create nel mio clusterper visualizzarne i dettagli.

Rimozione di oggetti di unità di condivisione dati dalle unità di condivisione dati

È possibile rimuovere uno o più oggetti da un'unità di condivisione dati utilizzando la procedura seguente.

Per rimuovere uno o più oggetti da una unità di condivisione dati

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati.
4. Nella sezione Unità di condivisione dati create nel mio account, scegliere Connetti al database. Per ulteriori informazioni, consulta Connessione a un database (anteprima).
5. Scegliere l'unità di condivisione dati che si desidera modificare, quindi selezionare Modifica. Viene visualizzata la pagina dei dettagli dell'unità di condivisione dati.
6. Per rimuovere uno o più oggetti di unità di condivisione dati dall'unità di condivisione dati, procedere in uno dei seguenti modi:

   - Per rimuovere gli schemi dall'unità di condivisione dati, scegliere uno o più schemi. Quindi scegliere Rimuovi. Amazon Redshift rimuove gli schemi specificati e tutti gli oggetti degli schemi specificati dall'unità di condivisione dati.

- Per rimuovere tabelle e viste dall'unità di condivisione dati, scegliere una o più tabelle e viste. Quindi scegliere Rimuovi. In alternativa, scegliere Rimuovi dallo schema per rimuovere tutte le tabelle e tutte le viste negli schemi specificati.
- Per rimuovere le funzioni definite dall'utente dall'unità di condivisione dati, scegliere una o più funzioni definite dall'utente. Quindi scegliere Rimuovi. In alternativa, scegliere Rimuovi dallo schema per rimuovere tutte le funzioni definite dall'utente negli schemi specificati.

Rimozione dei consumer di dati dalle unità di condivisione dati

È possibile rimuovere uno o più consumer di dati da una unità di condivisione dati. I consumatori di dati possono essere namespace di cluster che identificano in modo univoco cluster o account Amazon Redshift. AWS

Scegli uno o più consumatori di dati dagli ID o dall'account dello spazio dei nomi del cluster, quindi scegli Rimuovi. AWS

Amazon Redshift rimuove i consumer di dati specificati dall'unità di condivisione dati. L'accesso all'unità di condivisione dati si perde immediatamente.

Modifica delle unità di condivisione dati create nell'account

Modifica le unità di condivisione dati create nell'account mediante la console. Connettersi prima a un database per visualizzare l'elenco delle unità di condivisione dati create nell'account.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati.
4. Nella sezione Unità di condivisione dati create nel mio account scegli Connetti al database.
5. Scegliere l'unità di condivisione dati che si desidera modificare, quindi selezionare Modifica. Viene visualizzata la pagina dei dettagli dell'unità di condivisione dati.
6. Apportare eventuali modifiche nella finestra di dialogoOggetti unità di condivisione dati o nella sezione Consumer di dati.

> **Note**
>
> Se hai scelto di pubblicare il tuo datashare su AWS Glue Data Catalog, non puoi modificare la configurazione per pubblicare il datashare su altri account Amazon Redshift.

7. Seleziona Salvataggio delle modifiche.

Amazon Redshift aggiorna l'unità di condivisione dati con le modifiche.

Eliminazione delle unità di condivisione dati create nell'account

Eliminare le unità di condivisione dati create nell'account mediante la console. Connettersi prima a un database per visualizzare l'elenco delle unità di condivisione dati create nell'account.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati. Viene visualizzato un elenco di unità di condivisione dati.
4. Nella sezione Unità di condivisione dati create nel mio account scegli Connetti al database.
5. Scegliere una o più unità di condivisione dati da eliminare, quindi scegliere Elimina. Viene visualizzata la pagina Elimina unità di condivisione dati.

   L'eliminazione di un'unità di condivisione dati condivisa con Lake Formation non rimuove automaticamente le autorizzazioni associate in Lake Formation. Per rimuoverle, vai alla console di Lake Formation.
6. Digitare Elimina per confermare l'eliminazione delle unità di condivisione dati specificate.
7. Scegli Elimina.

Una volta eliminate le unità di condivisione dati, i consumer delle unità di condivisione dati perderanno l'accesso alle unità.

Esecuzione di query sulle unità di condivisione dati (anteprima)

Questa è una documentazione di pre-rilascio per le operazioni di scrittura in più data warehouse tramite la funzionalità di condivisione dei dati per Amazon Redshift, disponibile in anteprima

pubblica nella traccia PREVIEW_2023. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni di anteprima, vedere Partecipazione al servizio beta in Termini del servizio AWS.

Per ulteriori informazioni su come iniziare a condividere i dati, vai a Condivisione, scrittura, accesso ai dati (anteprima).

Creazione di database da unità di condivisione dati

Per iniziare a interrogare i dati nell'unità di condivisione dati, creare database da una unità. È possibile creare un solo database da una unità di condivisione dati specificata.

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere il cluster. Viene visualizzata la pagina dei dettagli del cluster.
3. Scegliere Unità di condivisione dati. Viene visualizzato un elenco di unità di condivisione dati.
4. Nella sezione Unità di condivisione dati da altri cluster, scegliere Connetti al database. Per ulteriori informazioni, consulta Connessione a un database (anteprima).
5. Scegliere una unità di condivisione dati da cui si desidera creare i database, quindi selezionare Crea database da unità di condivisione dati. Viene visualizzata la pagina Crea database da unità di condivisione dati.
6. In Nome del database, specificare un nome per il database. Il nome del database deve contenere un numero di caratteri alfanumerici (solo minuscoli) compreso tra 1 e 64 caratteri e non può essere una parola riservata.
7. Scegli Crea.

Dopo la creazione del database, puoi eseguire le query sui dati contenuti nel database oppure le operazioni di scrittura, se sono state assegnate, autorizzate e associate dall'amministratore del consumer.

Importazione e query di dati semistrutturati in Amazon Redshift

Usando il supporto di dati semistrutturati in Amazon Redshift, è possibile importare ed archiviare dati semistrutturati nei data warehouse di Amazon Redshift. Utilizzando il tipo di dati SUPER e il linguaggio PartiQL, Amazon Redshift espande la capacità di data warehouse per integrarsi con le origini dati SQL e NoSQL. In questo modo, Amazon Redshift consente un'analisi efficiente sui dati archiviati relazionali e semistrutturati come JSON.

Amazon Redshift offre due forme di supporto per i dati semistrutturati: il tipo di dati SUPER e Amazon Redshift Spectrum.

Utilizzare il tipo di dati SUPER se è necessario inserire o aggiornare piccoli batch di dati JSON con bassa latenza. Inoltre, utilizzare SUPER quando la query richiede coerenza elevata, prestazioni di query prevedibili, supporto di query complesso e facilità d'uso con schemi in evoluzione e dati senza schemi.

Al contrario, usa Amazon Redshift Spectrum con un formato di file aperto se la tua query sui dati richiede l'integrazione con AWS altri servizi e con dati archiviati principalmente in Amazon S3 per scopi di archiviazione.

Casi d'uso per il tipo di dati SUPER

Il supporto dei dati semistrutturati con il tipo di dati SUPER in Amazon Redshift offre prestazioni superiori, flessibilità e facilità d'uso. I seguenti casi d'uso dimostrano l'utilizzo del supporto di dati semistrutturati con SUPER.

Inserimento rapido e flessibile dei dati JSON: Amazon Redshift supporta transazioni rapide in grado di analizzare JSON e memorizzarlo come valore SUPER. Le transazioni di inserimento sono fino a cinque volte più veloci rispetto all'esecuzione degli stessi inserimenti in tabelle che hanno ridotto gli attributi di SUPER in colonne convenzionali. Ad esempio, si supponga che il JSON in arrivo sia nella forma {“a”:..., “b”:..., “c”“..., ...}. È possibile accelerare le prestazioni dell'inserimento molte volte memorizzando il JSON in entrata in una tabella TJ con una singola colonna SUPER S invece di memorizzarlo in una tabella TR convenzionale con colonne "a', 'b', "c' e così via. Quando ci sono centinaia di attributi nel JSON, il vantaggio in termini di prestazioni del tipo di dati SUPER diventa sostanziale.

Inoltre, il tipo di dati SUPER non ha bisogno di uno schema regolare. Non è necessario analizzare e ripulire il JSON in arrivo prima di archiviarlo. Ad esempio, si supponga che un JSON in entrata abbia un attributo stringa "c" e altri abbiano un attributo intero "c" senza il tipo di dati SUPER. In questo caso, è necessario separare le colonne c_string e c_int o pulire i dati. Al contrario, con il tipo di dati SUPER, tutti i dati JSON vengono archiviati durante l'importazione senza alcuna perdita di informazioni. Successivamente, è possibile utilizzare l'estensione PartiQL di SQL per analizzare le informazioni.

Query flessibili per il rilevamento: dopo aver archiviato i dati semistrutturati (ad esempio JSON) in un valore di dati SUPER, è possibile eseguire una query sui dati senza imporre uno schema. È possibile utilizzare la tipizzazione dinamica PartiQL e la sintassi permissiva per eseguire le query e scoprire i dati profondamente annidati necessari, senza la necessità di imporre uno schema prima della query.

Query flessibili per operazioni di estrazione, caricamento, trasformazione (ETL) in viste materializzate convenzionali: dopo aver archiviato i dati senza schema e semistrutturati in SUPER, è possibile utilizzare le viste materializzate PartiQL per analizzare i dati e suddividerli in viste materializzate.

Le viste materializzate con i dati ridotti sono un buon esempio di vantaggi in termini di prestazioni e usabilità per i casi di analisi classici. Quando si esegue un'analisi sui dati ridotti, l'organizzazione a colonne delle viste materializzate di Amazon Redshift offre prestazioni migliori. Inoltre, gli utenti e gli strumenti di Business Intelligence (BI) che richiedono uno schema convenzionale per i dati importati possono utilizzare viste (materializzate o virtuali) come presentazione convenzionale dello schema dei dati.

Dopo che le viste materializzate PartiQL hanno estratto i dati trovati in JSON o SUPER in viste materializzate a colonne convenzionali, è possibile eseguire una query sulle viste materializzate. Per ulteriori informazioni sul funzionamento del tipo di dati SUPER con le viste materializzate, consultare Utilizzo del tipo di dati SUPER con viste materializzate.

È possibile applicare le politiche di mascheramento dei dati dinamici ai valori `scalar` sui percorsi delle colonne di tipo SUPER. Per ulteriori informazioni sul mascheramento dei dati dinamici, consulta Mascheramento dinamico dei dati. Per informazioni su come usare il mascheramento dei dati dinamici con il tipo di dati SUPER, consulta Utilizzo del mascheramento dei dati dinamici con percorsi di tipo di dati SUPER (anteprima).

Per informazioni sull'utilizzo del tipo di dati SUPER, consultare Tipo SUPER.

Per esempi di utilizzo del tipo di dati SUPER, consulta le sotto sezioni di questo argomento che iniziano con: Set di dati di esempio SUPER.

Concetti per l'utilizzo del tipo di dati SUPER

Di seguito, sono riportati alcuni concetti sul tipo di dati SUPER di Amazon Redshift.

Capire qual è il tipo di dati SUPER in Amazon Redshift: il tipo di dati SUPER è un tipo di dati Amazon Redshift che consente l'archiviazione di array e strutture senza schemi che contengono dati scalari Amazon Redshift ed eventualmente array e strutture nidificate. Il tipo di dati SUPER può memorizzare in modo nativo diversi formati di dati semistrutturati, come JSON o dati provenienti da fonti orientate ai documenti. È possibile aggiungere una nuova colonna SUPER per memorizzare dati semistrutturati e scrivere query che accedono alla colonna SUPER, insieme alle solite colonne scalari. Per ulteriori informazioni sui tipi di fati SUPER, consultare Tipo SUPER.

Importare JSON senza schema in SUPER: con il il tipo di dati SUPER semistrutturati flessibile, Amazon Redshift può ricevere e importare JSON senza schema in un valore SUPER. Ad esempio, Amazon Redshift può inserire il valore JSON [10.5, "first"] in un valore SUPER [10.5, 'first'], ovvero una matrice contenente il decimale Amazon Redshift 10.5 e varchar 'first'. Amazon Redshift può importare il JSON in un valore SUPER utilizzando il comando COPY o la funzione di analisi JSON, come json_parse('[10.5, "first"]'). Sia COPY che il JSON di importazione `json_parse` utilizzano la sintassi di analisi rigorosa per impostazione predefinita. È inoltre possibile costruire valori SUPER che includono array e strutture utilizzando i dati stessi del database.

La colonna SUPER non richiede modifiche allo schema durante l'importazione delle strutture irregolari di JSON senza schema. Ad esempio, durante l'analisi di un click-stream, inizialmente nella colonna SUPER vengono archiviati strutture di "click" con gli attributi "IP" e "time". È possibile aggiungere un attributo "customer id" senza modificare lo schema al fine di importare tali modifiche.

Il formato nativo utilizzato per il tipo di dati SUPER è un formato binario che richiede meno spazio rispetto al valore JSON nella sua forma testuale. Ciò consente un'importazione più veloce e l'elaborazione in runtime dei valori SUPER sulla query.

Interroga i dati SUPER con PartiQL — PartiQL è un'estensione retrocompatibile di SQL-92 attualmente utilizzata da molti servizi. AWS Con l'uso di PartiQL, i costrutti SQL familiari combinano perfettamente l'accesso sia ai classici dati SQL tabulari che ai dati semistrutturati di SUPER. È possibile eseguire la navigazione di oggetti e array e annullare la nidificazione degli array. PartiQL estende il linguaggio SQL standard per esprimere dichiarativamente ed elaborare dati nidificati e multivalore.

PartiQL è un'estensione di SQL in cui i dati nidificati e senza schema delle colonne SUPER sono elementi prioritari. PartiQL non richiede che tutte le espressioni di query vengano controllate dal

tipo durante il tempo di compilazione della query. Questo approccio abilita espressioni di query che contengono il tipo di dati SUPER durante l'esecuzione della query quando si accede ai tipi effettivi dei dati all'interno delle colonne SUPER. Inoltre, PartiQL opera in una modalità permissiva in cui le incongruenze di tipo non causano errori ma restituiscono null. La combinazione del senza schema e dell'elaborazione di query permissiva rende PartiQL ideale per applicazioni di estrazione, caricamento, trasferimento (ELT) in cui la query SQL valuta i dati JSON che vengono importati nelle colonne SUPER.

Integrazione con Redshift Spectrum: Amazon Redshift supporta più aspetti di PartiQL durante l'esecuzione di query Redshift Spectrum su JSON, Parquet e altri formati con dati nidificati. Redshift Spectrum supporta solo i dati nidificati con schemi. Ad esempio, con Redshift Spectrum è possibile dichiarare che i dati JSON hanno un attributo nested_schemaful_esempio in uno schema ARRAY<STRUCT<a:INTEGER, b:DECIMAL(5,2)>>. Lo schema di questo attributo determina che i dati contengono sempre un array che a sua volta contiene una struttura con numero intero a e decimale b. Se i dati vengono modificati in modo da includere più attributi, viene modificato anche il tipo. Al contrario, il tipo di dati SUPER non richiede alcuno schema. È possibile memorizzare array con elementi della struttura che hanno attributi o tipi diversi. Inoltre, alcuni valori possono essere archiviati all'esterno degli array.

Per informazioni sulle funzioni che supportano il tipo di dati SUPER, consultare:

- Funzione ABS
- Funzione CEILING (oppure CEIL)
- Funzione FLOOR
- Funzione ROUND
- Funzione SIGN
- Funzione TRUNC

Considerazioni sui dati SUPER

Quando si utilizzano i dati SUPER, tenere in considerazione quanto riportato di seguito:

- Utilizzare il driver JDBC versione 1.2.50, il driver ODBC versione 1.4.17 o o successiva e il driver Amazon Redshift Python versione 2.0.872 o successiva.

  Per informazioni sui driver JDBC, consultare Configurazione di una connessione JDBC.

Per informazioni sui driver ODBC, consultare Configurazione di una connessione ODBC.

- Esempi di schema utilizzati nei seguenti argomenti sono disponibili in Set di dati di esempio SUPER.
- Tutti gli esempi di codice SQL utilizzati negli argomenti seguenti sono inclusi con lo stesso prefisso S3 per il download. Questi includono le istruzioni DDL (Data Definition Language) e COPY e alcune query modificate TPC-H che funzionano con SUPER.

  Per visualizzare o scaricare i file SQL, completare una delle operazioni seguenti:

  - Scaricare il file SQL del tutorial SUPER e il file TPC-H.
  - Utilizzando la CLI di Amazon S3, emettere il comando riportato di seguito. È possibile utilizzare il proprio percorso di destinazione.

```
aws s3 cp s3://redshift-downloads/semistructured/tutorialscripts/semistructured-
tutorial.sql /target/path
aws s3 cp s3://redshift-downloads/semistructured/tutorialscripts/
super_tpch_queries.sql /target/path
```

Per ulteriori informazioni sulle configurazioni SUPER, consultare Configurazioni di SUPER.

Set di dati di esempio SUPER

Lo schema di tabella e il modello di dati utilizzati per l'importazione e gli esempi di query sono definiti come segue.

```
/*customer-orders-lineitem*/
CREATE TABLE customer_orders_lineitem
(c_custkey bigint
,c_name varchar
,c_address varchar
,c_nationkey smallint
,c_phone varchar
,c_acctbal decimal(12,2)
,c_mktsegment varchar
,c_comment varchar
,c_orders super
);

/* Datamodel of documents to be stored in c_orders Super column would be as follows*/
```

```
ARRAY < STRUCT < o_orderkey:bigint
                           ,o_orderstatus:string
                           ,o_totalprice:double
                           ,o_orderdate:string
                           ,o_orderpriority:string
                           ,o_clerk:string
                           ,o_shippriority:int
                           ,o_comment:string
                           ,o_lineitems:ARRAY < STRUCT < l_partkey:bigint
                                                        ,l_suppkey:bigint
                                                        ,l_linenumber:int
                                                        ,l_quantity:double
                                                        ,l_extendedprice:double
                                                        ,l_discount:double
                                                        ,l_tax:double
                                                        ,l_returnflag:string
                                                        ,l_linestatus:string
                                                        ,l_shipdate:string
                                                        ,l_commitdate:string
                                                        ,l_receiptdate:string
                                                        ,l_shipinstruct:string
                                                        ,l_shipmode:string
                                                        ,l_comment:string
                                                        > >
                            > >

/*part*/
CREATE TABLE part
(
 p_partkey bigint
 ,p_name varchar
 ,p_mfgr varchar
 ,p_brand varchar
 ,p_type varchar
 ,p_size int
 ,p_container varchar
 ,p_retailprice decimal(12,2)
 ,p_comment varchar
 );

/*region-nations*/
CREATE TABLE region_nations
(
 r_regionkey smallint
```

```
 ,r_name varchar
 ,r_comment varchar
 ,r_nations super
 );

 /* Datamodel of documents to be stored in r_nations Super column would be as follows*/
 ARRAY < STRUCT < n_nationkey:int,n_name:string,n_comment:string > >

/*supplier-partsupp*/
CREATE TABLE supplier_partsupp
(
 s_suppkey bigint
 ,s_name varchar
 ,s_address varchar
 ,s_nationkey smallint
 ,s_phone varchar
 ,s_acctbal double precision
 ,s_comment varchar
 ,s_partsupps super
 );

 /* Datamodel of documents to be stored in s_partsupps Super column would be as
 follows*/
 ARRAY < STRUCT <
 ps_partkey:bigint,ps_availqty:int,ps_supplycost:double,ps_comment:string > >
```

Caricamento di dati semistrutturati in Amazon Redshift

Utilizzare il tipo di dati SUPER per mantenere ed eseguire query sui dati gerarchici e generici in Amazon Redshift. Amazon Redshift presenta la funzione `json_parse` per analizzare i dati in formato JSON e convertirli nella rappresentazione SUPER. Amazon Redshift supporta anche il caricamento delle colonne SUPER tramite il comando COPY. I formati file supportati sono JSON, Avro, text, formato CSV (comma-separated value, valori delimitati da virgole), Parquet e ORC.

Per informazioni sulle tabelle utilizzate negli esempi seguenti, consulta Set di dati di esempio SUPER.

Per ulteriori informazioni sulla funzione `json_parse`, consultare Funzione JSON_PARSE.

La codifica predefinita per il tipo di dati SUPER è ZSTD.

Analisi dei documenti JSON nelle colonne SUPER

È possibile inserire o aggiornare i dati JSON in una colonna SUPER utilizzando la funzione `json_parse`. La funzione analizza i dati in formato JSON e li converte nel tipo di dati SUPER, che è possibile utilizzare nelle istruzioni INSERT o UPDATE.

Nell'esempio seguente i dati JSON vengono inseriti in una colonna SUPER. Se la funzione `json_parse` non è presente nella query, Amazon Redshift considera il valore come una singola stringa anziché una stringa in formato JSON che deve essere analizzata.

Se si aggiorna una colonna di dati SUPER, Amazon Redshift richiede che il documento completo venga passato ai valori delle colonne. Amazon Redshift non supporta l'aggiornamento parziale.

```
INSERT INTO region_nations VALUES(0,
   'lar deposits. blithely final packages cajole. regular waters are final requests.
 regular accounts are according to',
   'AFRICA',
   JSON_PARSE('{"r_nations":[
      {"n_comment":" haggle. carefully final deposits detect slyly agai",
         "n_nationkey":0,
         "n_name":"ALGERIA"
      },
      {"n_comment":"ven packages wake quickly. regu",
         "n_nationkey":5,
         "n_name":"ETHIOPIA"
      },
      {"n_comment":" pending excuses haggle furiously deposits. pending, express pinto
 beans wake fluffily past t",
         "n_nationkey":14,
         "n_name":"KENYA"
      },
      {"n_comment":"rns. blithely bold courts among the closely regular packages use
 furiously bold platelets?",
         "n_nationkey":15,
         "n_name":"MOROCCO"
      },
      {"n_comment":"s. ironic, unusual asymptotes wake blithely r",
         "n_nationkey":16,
         "n_name":"MOZAMBIQUE"
      }
   ]
}'));
```

Utilizzo di COPY per caricare le colonne SUPER in Amazon Redshift

Nelle sezioni seguenti, è possibile ottenere informazioni sui diversi modi di utilizzare il comando COPY per caricare i dati JSON in Amazon Redshift.

Copia di dati da JSON e Avro

Utilizzando il supporto dei dati semistrutturati in Amazon Redshift, è possibile caricare un documento JSON senza ridurre gli attributi delle sue strutture JSON in più colonne.

Amazon Redshift fornisce due metodi per importare il documento JSON utilizzando COPY, anche con una struttura JSON completamente o parzialmente sconosciuta:

1. Archiviare i dati derivanti da un documento JSON in una singola colonna di dati SUPER utilizzando l'opzione `noshred`. Questo metodo è utile quando lo schema non è noto o si prevede che cambi. Pertanto, questo metodo rende più facile archiviare l'intera tupla in una singola colonna SUPER.
2. Ridurre il documento JSON in più colonne Amazon Redshift utilizzando l'opzione `auto` o `jsonpaths`. Gli attributi possono essere valori scalari di Amazon Redshift o valori SUPER.

È possibile utilizzare queste opzioni con i formati JSON o Avro.

La dimensione massima per un oggetto JSON prima della riduzione è 4 MB.

Copia di un documento JSON in una singola colonna di dati SUPER

Per copiare un documento JSON in una singola colonna di dati SUPER, creare una tabella con una singola colonna di dati SUPER.

```
CREATE TABLE region_nations_noshred (rdata SUPER);
```

Copiare i dati da Amazon S3 nella singola colonna di dati SUPER. Per importare i dati di origine JSON in una singola colonna di dati SUPER, specificare l'opzione `noshred` nella clausola FORMAT JSON.

```
COPY region_nations_noshred FROM 's3://redshift-downloads/semistructured/tpch-nested/
data/json/region_nation'
REGION 'us-east-1' IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3'
FORMAT JSON 'noshred';
```

Dopo che COPY ha importato correttamente il JSON, la tabella conterrà una colonna `rdata` di dati SUPER con i dati dell'intero oggetto JSON. I dati importati mantengono tutte le proprietà della gerarchia JSON. Tuttavia, per un'elaborazione efficiente delle query, gli elementi secondari della struttura vengono convertiti in tipi scalari Amazon Redshift.

Utilizzare la seguente query per recuperare la stringa JSON originale.

```
SELECT rdata FROM region_nations_noshred;
```

Quando Amazon Redshift genera una colonna di dati SUPER, diventa accessibile utilizzando JDBC come stringa tramite la serializzazione JSON. Per ulteriori informazioni, consultare Serializzazione di JSON nidificato complesso.

Copia di un documento JSON in più colonne di dati SUPER

È possibile suddividere un documento JSON in più colonne che possono essere colonne di dati SUPER o tipi scalari Amazon Redshift. Amazon Redshift diffonde diverse parti dell'oggetto JSON su colonne diverse.

```
CREATE TABLE region_nations
(
 r_regionkey smallint
 ,r_name varchar
 ,r_comment varchar
 ,r_nations super
 );
```

Per copiare i dati dell'esempio precedente nella tabella, specificare l'opzione AUTO nella clausola FORMAT JSON per dividere il valore JSON su più colonne. COPY corrisponde agli attributi JSON di primo livello con i nomi delle colonne e consente l'inserimento dei valori nidificati come valori SUPER, ad esempio array e oggetti JSON.

```
COPY region_nations FROM 's3://redshift-downloads/semistructured/tpch-nested/data/json/
region_nation'
REGION 'us-east-1' IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3'
FORMAT JSON 'auto';
```

Quando i nomi degli attributi JSON sono in lettere maiuscole e minuscole miste, specificare l'opzione `auto ignorecase` nella clausola FORMAT JSON. Per ulteriori informazioni sul comando COPY, consultare Caricamento da dati JSON utilizzando l'opzione 'auto ignorecase'.

In alcuni casi, c'è una mancata corrispondenza tra i nomi delle colonne e gli attributi JSON o l'attributo da caricare è nidificato con una profondità di più di un livello. In tal caso, utilizzare un file `jsonpaths` per mappare manualmente gli attributi JSON alle colonne Amazon Redshift.

```
CREATE TABLE nations
(
 regionkey smallint
 ,name varchar
 ,comment super
 ,nations super
 );
```

Si supponga di voler caricare i dati in una tabella in cui i nomi delle colonne non corrispondono agli attributi JSON. Nell'esempio seguente, la tabella `nations` è una tabella del genere. È possibile creare un file `jsonpaths` che mappa i percorsi degli attributi alle colonne della tabella in base alla loro posizione nell'array `jsonpaths`.

```
{"jsonpaths": [
       "$.r_regionkey",
       "$.r_name",
       "$.r_comment",
       "$.r_nations
    ]
}
```

La posizione del file `jsonpaths` viene utilizzata come argomento per FORMAT JSON.

```
COPY nations FROM 's3://redshift-downloads/semistructured/tpch-nested/data/json/
region_nation'
REGION 'us-east-1' IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3'
FORMAT JSON 's3://redshift-downloads/semistructured/tpch-nested/data/jsonpaths/
nations_jsonpaths.json';
```

Utilizzare la query seguente per accedere alla tabella che mostra la distribuzione dei dati su più colonne. Le colonne di dati SUPER vengono stampate utilizzando il formato JSON.

```
SELECT r_regionkey,r_name,r_comment,r_nations[0].n_nationkey FROM region_nations ORDER
 BY 1,2,3 LIMIT 1;
```

I file Jsonpath mappano i campi del documento JSON alle colonne della tabella. È possibile estrarre colonne aggiuntive, come le chiavi di distribuzione e ordinamento, mentre continuano a caricare il documento completo come colonna SUPER. La seguente query carica il documento completo nella colonna nazioni. La colonna `name` è la chiave di ordinamento e la colonna `regionkey` è la chiave di distribuzione.

```
CREATE TABLE nations_sorted (
    regionkey smallint,
    name varchar,
    nations super
) DISTKEY(regionkey) SORTKEY(name);
```

Il root jsonpath «$» mappa alla radice del documento come segue:

```
{"jsonpaths": [
       "$.r_regionkey",
       "$.r_name",
       "$"
    ]
}
```

La posizione del file jsonpath viene utilizzata come argomento per FORMAT JSON.

```
COPY nations_sorted FROM 's3://redshift-downloads/semistructured/tpch-nested/data/json/
region_nation'
REGION 'us-east-1' IAM_ROLE 'arn:aws:iam::xxxxxxxxxxx:role/Redshift-S3'
FORMAT JSON 's3://redshift-downloads/semistructured/tpch-nested/data/jsonpaths/
nations_sorted_jsonpaths.json';
```

Copia di dati da testo e CSV

Amazon Redshift rappresenta colonne SUPER nei formati testo e CSV come JSON serializzato. È necessaria una formattazione JSON valida per il caricamento delle colonne SUPER con le informazioni sul tipo corrette. Oggetti, array, numeri, booleani e valori nulli non quotati. Avvolgi i valori di stringa tra virgolette. Le colonne SUPER utilizzano regole di escape standard per i formati testo e CSV. Per CSV, i delimitatori sono sfuggiti secondo lo standard CSV. Per il testo, se il delimitatore scelto appare anche in un campo SUPER, utilizzare l'opzione ESCAPE durante COPY e UNLOAD

```
COPY region_nations FROM 's3://redshift-downloads/semistructured/tpch-nested/data/csv/
region_nation'
```

```
REGION 'us-east-1' IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3'
FORMAT CSV;
```

```
COPY region_nations FROM 's3://redshift-downloads/semistructured/tpch-nested/data/text/
region_nation'
REGION 'us-east-1' IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3'
DELIMITER ','
ESCAPE;
```

Copia dei dati da Parquet e ORC in formato colonna

Se i dati semistrutturati o nidificati sono già disponibili in formato Apache Parquet o Apache ORC, è possibile usare il comando COPY per inserire i dati in Amazon Redshift.

La struttura della tabella Amazon Redshift deve corrispondere al numero di colonne e ai tipi di dati delle colonne dei file Parquet o ORC. Specificando SERIALIZETOJSON nel comando COPY, è possibile caricare qualsiasi tipo di colonna nel file che si allinea con una colonna SUPER nella tabella come SUPER. Ciò include la struttura e i tipi di array.

```
COPY region_nations FROM 's3://redshift-downloads/semistructured/tpch-nested/data/
parquet/region_nation'
REGION 'us-east-1' IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3'
FORMAT PARQUET SERIALIZETOJSON;
```

Nell'esempio seguente viene utilizzato un formato ORC.

```
COPY region_nations FROM 's3://redshift-downloads/semistructured/tpch-nested/data/orc/
region_nation'
IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3'
FORMAT ORC SERIALIZETOJSON;
```

Quando gli attributi dei tipi di dati di data o ora sono in ORC, Amazon Redshift li converte in varchar dopo averli codificati in SUPER.

Scaricamento dei dati semistrutturati

Puoi scaricare tabelle con colonne dati SUPER su Amazon S3 in diversi formati.

Argomenti

Scaricamento di dati semistrutturati in formato CSV o testo

È possibile scaricare tabelle con colonne di dati SUPER su Amazon S3 in formato CSV (comma-separated value, valori delimitati da virgole) o testo. Utilizzando una combinazione di clausole di navigazione e annullamento della nidificazione, Amazon Redshift scarica i dati gerarchici in formato dati SUPER su Amazon S3 in formato CSV o testo. Successivamente, è possibile creare tabelle esterne contro i dati scaricati ed eseguire una query utilizzando Redshift Spectrum. Per informazioni sull'utilizzo di UNLOAD e delle autorizzazioni IAM richieste, consultare UNLOAD.

Prima di eseguire l'esempio seguente, compila la tabella region_nations utilizzando i processi descritti in Caricamento di dati semistrutturati in Amazon Redshift. Per informazioni sulle tabelle utilizzate nell'esempio seguente, consulta Set di dati di esempio SUPER.

Nell'esempio seguente i dati sono scaricati in Amazon S3.

```
UNLOAD ('SELECT * FROM region_nations')
TO 's3://xxxxxx/'
IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3-Write'
DELIMITER AS '|'
GZIP
ALLOWOVERWRITE;
```

A differenza di altri tipi di dati in cui una stringa definita dall'utente rappresenta un valore nullo, Amazon Redshift esporta le colonne di dati SUPER utilizzando il formato JSON e li rappresenta come null come determinato dal formato JSON. Di conseguenza, le colonne di dati SUPER ignorano l'opzione NULL [AS] utilizzata nei comandi UNLOAD.

Scaricamento di dati semistrutturati nel formato Parquet

Puoi scaricare tabelle con colonne dati SUPER su Amazon S3 nel formato Parquet. Amazon Redshift rappresenta le colonne SUPER in Parquet come tipo di dati JSON. Ciò consente di rappresentare i dati semistrutturati in Parquet. È possibile eseguire query su queste colonne utilizzando Redshift Spectrum o reinserirle in Amazon Redshift utilizzando il comando COPY. Per informazioni sull'utilizzo di UNLOAD e delle autorizzazioni IAM richieste, consultare UNLOAD.

Nell'esempio seguente i dati sono scaricati in Amazon S3 nel formato Parquet.

```
UNLOAD ('SELECT * FROM region_nations')
TO 's3://xxxxxx/'
IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3-Write'
FORMAT PARQUET;
```

Query sui dati semistrutturati

Amazon Redshift utilizza il linguaggio PartiQL per offrire accesso compatibile con SQL a dati relazionali, semistrutturati e nidificati.

PartiQL funziona con tipi dinamici. Questo approccio consente di filtrare, unire e aggregare intuitivi sulla combinazione di set di dati strutturati, semistrutturati e nidificati. La sintassi PartiQL utilizza la notazione puntata e l'indice di array per la navigazione dei percorsi quando si accede ai dati nidificati. Consente inoltre agli elementi della clausola FROM di iterare su array e uso per le operazioni di annullamento nidificazione. Seguendo, è possibile trovare descrizioni di diversi modelli di query che combinano l'uso del tipo di dati SUPER con percorsi e array di navigazione, annullamento, nidificazione e join.

Per informazioni sulle tabelle utilizzate nell'esempio seguente, consulta Set di dati di esempio SUPER.

Navigazione

Amazon Redshift utilizza PartiQL per abilitare la navigazione in array e strutture utilizzando rispettivamente la parentesi [...] e la notazione a punti. Inoltre, è possibile combinare la navigazione in strutture utilizzando la notazione a punti e gli array utilizzando la notazione con parentesi. Ad esempio, nell'esempio seguente si assume che la colonna di dati SUPER `c_orders` sia un array con una struttura e che un attributo sia denominato `o_orderkey`.

Per acquisire dati nella tabella `customer_orders_lineitem`, eseguire il seguente comando. Sostituire il ruolo IAM con le proprie credenziali.

```
COPY customer_orders_lineitem FROM 's3://redshift-downloads/semistructured/tpch-nested/
data/json/customer_orders_lineitem'
REGION 'us-east-1' IAM_ROLE 'arn:aws:iam::xxxxxxxxxxxx:role/Redshift-S3'
FORMAT JSON 'auto';

SELECT c_orders[0].o_orderkey FROM customer_orders_lineitem;
```

Amazon Redshift utilizza anche un alias della tabella come prefisso alla notazione. L'esempio seguente è la stessa query dell'esempio precedente.

```
SELECT cust.c_orders[0].o_orderkey FROM customer_orders_lineitem AS cust;
```

È possibile utilizzare le notazioni con punti e parentesi in tutti i tipi di query, ad esempio filtraggio, join e aggregazione. È possibile utilizzare queste notazioni in una query in cui ci sono normalmente riferimenti di colonna. Nell'esempio seguente viene utilizzata un'istruzione SELECT che filtra i risultati.

```
SELECT count(*) FROM customer_orders_lineitem WHERE c_orders[0]. o_orderkey IS NOT
 NULL;
```

Nell'esempio seguente viene utilizzata la navigazione con parentesi e punti nelle clausole GROUP BY e ORDER BY.

```
SELECT c_orders[0].o_orderdate,
       c_orders[0].o_orderstatus,
       count(*)
FROM customer_orders_lineitem
WHERE c_orders[0].o_orderkey IS NOT NULL
GROUP BY c_orders[0].o_orderstatus,
         c_orders[0].o_orderdate
ORDER BY c_orders[0].o_orderdate;
```

Annullamento di query

Per annullare le query, Amazon Redshift utilizza la sintassi PartiQL per eseguire l'iterazione su array SUPER. Lo fa navigando nell'array utilizzando la clausola FROM di una query. Utilizzando l'esempio precedente, nell'esempio seguente vengono eseguite interazioni sui valori dell'attributo per `c_orders`.

```
SELECT c.*, o FROM customer_orders_lineitem c, c.c_orders o;
```

La sintassi di annullamento nidificazione è un'estensione della clausola FROM. In SQL standard, la clausola FROM `x (AS) y` significa che in `y` vengono eseguite iterazioni su ogni tupla in relazione `x`. In questo caso, `x` si riferisce a una relazione e `y` si riferisce a un alias per relazione `x`. Analogamente, la sintassi PartiQL di unnesting utilizzando l'elemento della clausola FROM `x (AS) y` significa che in `y` vengono eseguite iterazioni su ciascun valore (SUPER) nell'espressione di array (SUPER) `x`. In questo caso, `x` è un'espressione SUPER e `y` è un alias per `x`.

Per la navigazione regolare, con l'operando sinistro si può anche utilizzare la notazione con punti e parentesi. Nell'esempio precedente, `customer_orders_lineitem c` è l'iterazione sulla tabella di base `customer_order_lineitem` e `c.c_orders o` è l'iterazione sull'array `c.c_orders`. Per eseguire iterazioni sull'attributo `o_lineitems`, che è un array all'interno di un array, si aggiungono più clausole.

```
SELECT c.*, o, l FROM customer_orders_lineitem c, c.c_orders o, o.o_lineitems l;
```

Amazon Redshift supporta anche un indice dell'array quando si esegue l'iterazione sull'array utilizzando la parola chiave AT. Nella clausola `x AS y AT z` vengono eseguite iterazioni sull'array `x` e genera il campo `z`, che è l'indice dell'array. Nell'esempio seguente viene illustrato il funzionamento di un indice di array:

```
SELECT c_name,
       orders.o_orderkey AS orderkey,
       index AS orderkey_index
FROM customer_orders_lineitem c, c.c_orders AS orders AT index
ORDER BY orderkey_index;

c_name             | orderkey | orderkey_index
-------------------+----------+----------------
Customer#000008251 | 3020007  |        0
Customer#000009452 | 4043971  |        0
  (2 rows)
```

Nell'esempio seguente sono eseguite iterazioni su un array scalare.

```
CREATE TABLE bar AS SELECT json_parse('{"scalar_array": [1, 2.3, 45000000]}') AS data;

SELECT index, element FROM bar AS b, b.data.scalar_array AS element AT index;

 index | element
-------+----------
     0 | 1
     1 | 2.3
     2 | 45000000
(3 rows)
```

Nell'esempio seguente viene eseguita un'iterazione su un array di più livelli. L'esempio utilizza più clausole unnest per eseguire l'iterazione negli array più interni. Nell'array AS

`f.multi_level_array` viene eseguita un'iterazione su `multi_level_array`. L'elemento array AS è l'iterazione sugli array entro `multi_level_array`.

```
CREATE TABLE foo AS SELECT json_parse('[[1.1, 1.2], [2.1, 2.2], [3.1, 3.2]]') AS
 multi_level_array;

SELECT array, element FROM foo AS f, f.multi_level_array AS array, array AS element;

   array   | element
-----------+---------
 [1.1,1.2] | 1.1
 [1.1,1.2] | 1.2
 [2.1,2.2] | 2.1
 [2.1,2.2] | 2.2
 [3.1,3.2] | 3.1
 [3.1,3.2] | 3.2
(6 rows)
```

Per ulteriori informazioni sulla clausola FROM, consultare Clausola FROM.

Nidificazione di oggetti

Per eseguire la nidificazionet degli oggetti, Amazon Redshift utilizza la sintassi PartiQL per eseguire l'iterazione sugli oggetti SUPER. Lo fa utilizzando la clausola FROM di una query con la parola chiave UNPIVOT. In questo caso, l'espressione è l'oggetto. `c.c_orders[0]` La query di esempio esegue un'iterazione su ogni attributo restituito dall'oggetto.

```
SELECT attr as attribute_name, json_typeof(val) as value_type
FROM customer_orders_lineitem c, UNPIVOT c.c_orders[0] AS val AT attr
WHERE c_custkey = 9451;

 attribute_name  | value_type
-----------------+------------
 o_orderstatus   | string
 o_clerk         | string
 o_lineitems     | array
 o_orderdate     | string
 o_shippriority  | number
 o_totalprice    | number
 o_orderkey      | number
 o_comment       | string
 o_orderpriority | string
```

```
(9 rows)
```

Come con l'annullamento, la sintassi di nidificazione è un'estensione della clausola FROM. La differenza è che la sintassi di nidificazione utilizza la parola chiave UNPIVOT per indicare che sta iterando su un oggetto anziché su un array. Utilizza l'AS `value_alias` per l'iterazione su tutti i valori all'interno di un oggetto e utilizza l'AT `attribute_alias` per l'iterazione su tutti gli attributi. Considerate il seguente frammento di sintassi:

```
UNPIVOT expression AS value_alias [ AT attribute_alias ]
```

Amazon Redshift supporta l'utilizzo del unpivoting degli oggetti e del unnesting degli array in un'unica clausola FROM come segue:

```
SELECT attr as attribute_name, val as object_value
FROM customer_orders_lineitem c, c.c_orders AS o, UNPIVOT o AS val AT attr
WHERE c_custkey = 9451;
```

Quando utilizzi la nidificazione degli oggetti, Amazon Redshift non supporta la nidificazione correlata. In particolare, supponiamo di avere un caso in cui ci sono più esempi di nidificazione in diversi livelli di query e che la nidificazione interna faccia riferimento a quella esterno. Amazon Redshift non supporta questo tipo di nidificazione multipla.

Per ulteriori informazioni sulla clausola FROM, consultare Clausola FROM. Per gli esempi di query su dati strutturati con PIVOT e UNPIVOT consulta Esempi PIVOT e UNPIVOT.

Digitazione dinamica

La digitazione dinamica non richiede il casting esplicito dei dati estratti dai percorsi con punti e parentesi. Amazon Redshift utilizza la digitazione dinamica per elaborare dati SUPER senza schema senza la necessità di dichiarare i tipi di dati prima di utilizzarli nella query. La digitazione dinamica utilizza i risultati della navigazione nelle colonne di dati SUPER senza doverne eseguire esplicitamente il casting nei tipi Amazon Redshift. La digitazione dinamica è più utile nei join e nelle clausole GROUP BY. Nell'esempio seguente viene utilizzata un'istruzione SELECT che non richiede il casting esplicito delle espressioni con punti e parentesi ai normali tipi Amazon Redshift. Per informazioni sulla compatibilità e la conversione dei tipi, consultare Conversione e compatibilità dei tipi.

```
SELECT c_orders[0].o_orderkey
```

```
FROM customer_orders_lineitem
WHERE c_orders[0].o_orderstatus = 'P';
```

Il segno di uguaglianza in questa query restituisce `true`quando c_orders[0].o_orderstatus è la stringa 'P'. In tutti gli altri casi, il segno di uguaglianza restituisce`false`, compresi i casi in cui gli argomenti dell'uguaglianza sono tipi diversi.

Digitazione dinamica e statica

Senza utilizzare la digitazione dinamica, non è possibile determinare se c_orders[0].o_orderstatus è una stringa, un numero intero o una struttura. È possibile determinare solo che c_orders[0].o_orderstatus è un tipo di dati SUPER, che può essere uno scalare Amazon Redshift, un array o una struttura. Il tipo statico di c_orders[0].o_orderstatus è un tipo di dati SUPER. Convenzionalmente, in SQL un tipo è implicitamente un tipo statico.

Amazon Redshift utilizza la digitazione dinamica per l'elaborazione dei dati senza schema. Quando la query valuta i dati, c_orders[0].o_orderstatus diventa un tipo specifico. Ad esempio, la valutazione di c_orders[0].o_orderstatus sul primo record di customer_orders_lineitem può restituire un numero intero. La valutazione sul secondo record può restituire una stringa. Questi sono i tipi dinamici dell'espressione.

Se si utilizza un operatore o una funzione SQL con espressioni con punti e parentesi che hanno tipi dinamici, Amazon Redshift produce risultati simili all'utilizzo dell'operatore SQL standard o della funzione con i rispettivi tipi statici. In questo esempio, quando il tipo dinamico dell'espressione del percorso è una stringa, il confronto con la stringa 'P' è significativo. Ogni volta che il tipo dinamico di c_orders[0].o_orderstatus è qualsiasi altro tipo di dati che non sia stringa, l'uguaglianza restituisce false. Le altre funzioni restituiscono null quando vengono utilizzati argomenti errati.

L'esempio seguente scrive la query precedente con la digitazione statica:

```
SELECT c_custkey
FROM customer_orders_lineitem
WHERE CASE WHEN JSON_TYPEOF(c_orders[0].o_orderstatus) = 'string'
           THEN c_orders[0].o_orderstatus::VARCHAR = 'P'
           ELSE FALSE END;
```

Si noti la seguente distinzione tra predicati di uguaglianza e predicati di confronto. Nell'esempio precedente, se sostituisci il predicato di uguaglianza con un less-than-or-equal predicato, la semantica produce null anziché false.

```
SELECT c_orders[0]. o_orderkey
FROM customer_orders_lineitem
WHERE c_orders[0].o_orderstatus <= 'P';
```

In questo esempio, se c_orders[0].o_orderstatus è una stringa, Amazon Redshift restituisce true se è alfabeticamente uguale o inferiore a 'P'. Amazon Redshift restituisce false se è alfabeticamente più grande di 'P'. Tuttavia, se c_orders[0].o_orderstatus non è una stringa, Amazon Redshift restituisce null poiché non è in grado di confrontare valori di tipi diversi, come mostrato nella seguente query:

```
SELECT c_custkey
FROM customer_orders_lineitem
WHERE CASE WHEN JSON_TYPEOF(c_orders[0].o_orderstatus) = 'string'
           THEN c_orders[0].o_orderstatus::VARCHAR <= 'P'
           ELSE NULL END;
```

La digitazione dinamica non esclude dai confronti di tipi minimamente comparabili. Ad esempio, è possibile convertire entrambi i tipi scalari CHAR e VARCHAR di Amazon Redshift in SUPER. Sono paragonabili come stringhe, ad esempio come nel caso di ignorare i caratteri di spazi finali simili ai tipi CHAR e VARCHAR di Amazon Redshift. Allo stesso modo, numeri interi, decimali e a virgola mobile sono comparabili come valori SUPER. Per le colonne decimali in particolare, ogni valore può anche avere una scala diversa. Amazon Redshift li considera ancora come tipi dinamici.

Amazon Redshift supporta anche l'uguaglianza su oggetti e array valutati come deep equal, ad esempio la valutazione approfondita di oggetti o array e il confronto di tutti gli attributi. Utilizzare deep equal con cautela, perché il processo di esecuzione di deep equal può richiedere molto tempo.

Utilizzo della digitazione dinamica per i join

Per i join, la digitazione dinamica corrisponde automaticamente ai valori con diversi tipi dinamici senza eseguire un'analisi CASE WHEN lunga per scoprire quali tipi di dati possono apparire. Ad esempio, si supponga che l'organizzazione abbia modificato nel tempo il formato utilizzato per le chiavi di parte.

Le chiavi di parte integer iniziali emesse vengono sostituite da chiavi di parte string, come 'A55', e successivamente sostituite di nuovo da chiavi di parte array, come ['X', 10] combinando una stringa e un numero. Amazon Redshift non deve eseguire una analisi lunga dei casi sulle chiavi di parte e può utilizzare i join come mostrato nell'esempio seguente.

```
SELECT c.c_name
```

```
    ,l.l_extendedprice
    ,l.l_discount
FROM customer_orders_lineitem c
    ,c.c_orders o
    ,o.o_lineitems l
    ,supplier_partsupp s
    ,s.s_partsupps ps
WHERE l.l_partkey = ps.ps_partkey
AND c.c_nationkey = s.s_nationkey
ORDER BY c.c_name;
```

Nell'esempio seguente viene mostrato quanto sia complessa e inefficiente la stessa query se non viene utilizzata la digitazione dinamica:

```
SELECT c.c_name
    ,l.l_extendedprice
    ,l.l_discount
FROM customer_orders_lineitem c
    ,c.c_orders o
    ,o.o_lineitems l
    ,supplier_partsupp s
    ,s.s_partsupps ps
WHERE CASE WHEN IS_INTEGER(l.l_partkey) AND IS_INTEGER(ps.ps_partkey)
           THEN l.l_partkey::integer = ps.ps_partkey::integer
           WHEN IS_VARCHAR(l.l_partkey) AND IS_VARCHAR(ps.ps_partkey)
           THEN l.l_partkey::varchar = ps.ps_partkey::varchar
           WHEN IS_ARRAY(l.l_partkey) AND IS_ARRAY(ps.ps_partkey)
                AND IS_VARCHAR(l.l_partkey[0]) AND IS_VARCHAR(ps.ps_partkey[0])
                AND IS_INTEGER(l.l_partkey[1]) AND IS_INTEGER(ps.ps_partkey[1])
           THEN l.l_partkey[0]::varchar = ps.ps_partkey[0]::varchar
                AND l.l_partkey[1]::integer = ps.ps_partkey[1]::integer
           ELSE FALSE END
AND c.c_nationkey = s.s_nationkey
ORDER BY c.c_name;
```

Semantica permissiva

Per impostazione predefinita, le operazioni di navigazione sui valori SUPER restituiscono null invece di restituire un errore quando la navigazione non è valida. La navigazione tra gli oggetti non è valida se il valore SUPER non è un oggetto o se il valore SUPER è un oggetto ma non contiene il nome dell'attributo utilizzato nella query. Ad esempio, la seguente query accede a un nome attributo non valido nella colonna di dati SUPER cdata:

```
SELECT c.c_orders.something FROM customer_orders_lineitem c;
```

La navigazione nell'array restituisce null se il valore SUPER non è un array o se l'indice dell'array è fuori dai limiti. La query seguente restituisce null perché c_orders[1][1] è fuori dai limiti.

```
SELECT c.c_orders[1][1] FROM customer_orders_lineitem c;
```

La sintassi permissiva è particolarmente utile quando si utilizza la digitazione dinamica per eseguire il casting di un valore SUPER. Il casting di un valore SUPER sul tipo non corretto restituisce null invece che un errore se il casting non è valido. Ad esempio, la seguente query restituisce null perché non è possibile eseguire il casting del valore stringa 'Good' dell'attributo oggetto o_orderstatus su INTEGER. Amazon Redshift restituisce un errore per un casting da VARCHAR a INTEGER ma non per un casting SUPER.

```
SELECT c.c_orders.o_orderstatus::integer FROM customer_orders_lineitem c;
```

Tipi di introspezione

Le colonne di dati SUPER supportano funzioni di ispezione che restituiscono il tipo dinamico e altre informazioni di tipo sul valore SUPER. L'esempio più comune è la funzione scalare JSON_TYPEOF che restituisce un VARCHAR con valori booleani, number, string, object, array o null, a seconda del tipo dinamico del valore SUPER. Amazon Redshift supporta le seguenti funzioni booleane per le colonne di dati SUPER:

- DECIMAL_PRECISION
- DECIMAL_SCALE
- IS_ARRAY
- IS_BIGINT
- IS_CHAR
- IS_DECIMAL
- IS_FLOAT
- IS_INTEGER
- IS_OBJECT
- IS_SCALARE
- IS_SMALLINT

- IS_VARCHAR
- JSON_TYPEOF

Tutte queste funzioni restituiscono false se il valore di input è null. IS_SCALAR, IS_OBJECT e IS_ARRAY si escludono a vicenda e coprono tutti i valori possibili ad eccezione di null.

Per dedurre i tipi corrispondenti ai dati, Amazon Redshift utilizza la funzione JSON_TYPEOF che restituisce il tipo (il livello superiore) del valore SUPER, come mostrato nell'esempio seguente:

```
SELECT JSON_TYPEOF(r_nations) FROM region_nations;
 json_typeof
 -------------
 array
(1 row)
```

```
SELECT JSON_TYPEOF(r_nations[0].n_nationkey) FROM region_nations;
 json_typeof
 -------------
 number
```

Amazon Redshift lo vede come una singola stringa lunga, simile all'inserimento di questo valore in una colonna VARCHAR invece che in una SUPER. Poiché la colonna è SUPER, la singola stringa è ancora un valore SUPER valido e la differenza è annotata in JSON_TYPEOF:

```
SELECT IS_VARCHAR(r_nations[0].n_name) FROM region_nations;
 is_varchar
-------------
 true
(1 row)
```

```
SELECT r_nations[4].n_name FROM region_nations
WHERE CASE WHEN IS_INTEGER(r_nations[4].n_nationkey)
           THEN r_nations[4].n_nationkey::INTEGER = 15
           ELSE false END;
```

Order by (Ordina per)

Amazon Redshift non definisce i confronti SUPER tra valori con diversi tipi dinamici. Un valore SUPER che è una stringa non è né più piccolo né più grande di un valore SUPER che è un numero.

Per utilizzare le clausole ORDER BY con colonne SUPER, Amazon Redshift definisce un ordine totale tra diversi tipi da osservare quando Amazon Redshift classifica i valori SUPER utilizzando le clausole ORDER BY. L'ordine tra i tipi dinamici è booleano, numero, stringa, array, oggetto. Nell'esempio seguente vengono illustrati gli ordini di tipi diversi:

```
INSERT INTO region_nations VALUES
(100,'name1','comment1','AWS'),
(200,'name2','comment2',1),
(300,'name3','comment3',ARRAY(1, 'abc', null)),
(400,'name4','comment4',-2.5),
(500,'name5','comment5','Amazon');

SELECT r_nations FROM region_nations order by r_nations;

r_nations
----------------
 -2.5
 1
 "Amazon"
 "AWS"
 [1,"abc",null]
(5 rows)
```

Per ulteriori informazioni sulla clausola ORDER BY, consultare Clausola ORDER BY.

Operatori e funzioni

Amazon Redshift fornisce il seguente supporto funzione di operatori e funzioni SUPER.

Operatori aritmetici

I valori SUPER supportano tutti gli operatori aritmetici di base +, -, *,/,% con la digitazione dinamica. Il tipo risultante dell'operazione rimane SUPER. Per tutti gli operatori, ad eccezione dell'operatore binario +, gli operandi di input devono essere numeri. In caso contrario, Amazon Redshift restituisce null. La distinzione tra valori decimali e a virgola mobile viene mantenuta quando Amazon Redshift esegue questi operatori e il tipo dinamico non cambia. Tuttavia, la scala decimale cambia quando si utilizzano moltiplicazioni e divisioni. Gli overflow aritmetici causano ancora errori di query, non vengono modificati in null. L'operatore binario + esegue l'aggiunta se gli input sono numeri o la concatenazione se gli input sono una stringa. Se un operando è una stringa e l'altro operando è un

numero, il risultato è null. Gli operatori con prefisso unario + e - restituisce null se il valore SUPER non è un numero, come illustrato nell'esempio seguente:

```
SELECT (c_orders[0]. o_orderkey + 0.5) * c_orders[0]. o_orderkey / 10 AS math FROM
 customer_orders_lineitem;
           math
----------------------------
 1757958232200.1500
(1 row)
```

La digitazione dinamica consente ai valori decimali in SUPER di avere scale diverse. Amazon Redshift considera i valori decimali come se fossero tipi statici diversi e consente tutte le operazioni matematiche. Amazon Redshift calcola dinamicamente la scala risultante in base alle scale degli operandi. Se uno degli operandi è un numero a virgola mobile, Amazon Redshift promuove l'altro operando a un numero a virgola mobile e genera il risultato come numero a virgola mobile.

Funzioni aritmetiche

Amazon Redshift supporta le seguenti funzioni aritmetiche per le colonne SUPER. Restituiscono null se l'input non è un numero:

- FLOOR. Per ulteriori informazioni, consulta Funzione FLOOR.
- CEIL e CEILING. Per ulteriori informazioni, consulta Funzione CEILING (oppure CEIL).
- ROUND. Per ulteriori informazioni, consulta Funzione ROUND.
- TRUNC. Per ulteriori informazioni, consulta Funzione TRUNC.
- ABS. Per ulteriori informazioni, consulta Funzione ABS.

Nell'esempio seguente vengono utilizzate funzioni aritmetiche per eseguire query sui dati:

```
SELECT x, FLOOR(x), CEIL(x), ROUND(x)
FROM (
    SELECT (c_orders[0]. o_orderkey + 0.5) * c_orders[0].o_orderkey / 10 AS x
    FROM customer_orders_lineitem
    );

         x          |     floor     |     ceil      |     round
--------------------+---------------+---------------+---------------
 1389636795898.0500  | 1389636795898  | 1389636795899  | 1389636795898
```

La funzione ABS mantiene la scala del decimale di input mentre FLOOR, CEIL. ROUND elimina la scala del decimale di input.

Funzioni di array

Amazon Redshift supporta la composizione dell'array e le funzioni di utilità seguenti array, array_concat, subarray, array_flatten, get_array_length e split_to_array.

È possibile costruire array SUPER da valori nei tipi di dati Amazon Redshift utilizzando la funzione ARRAY, inclusi altri valori SUPER. Nell'esempio seguente viene utilizzata la funzione variadica ARRAY:

```
SELECT ARRAY(1, c.c_custkey, NULL, c.c_name, 'abc') FROM customer_orders_lineitem c;
                               array
 ------------------------------------------------------
[1,8401,null,""Customer#000008401"",""abc""]
[1,9452,null,""Customer#000009452"",""abc""]
[1,9451,null,""Customer#000009451"",""abc""]
[1,8251,null,""Customer#000008251"",""abc""]
[1,5851,null,""Customer#000005851"",""abc""]
(5 rows)
```

Nell'esempio seguente viene utilizzata la concatenazione di array con la funzione ARRAY_CONCAT:

```
SELECT ARRAY_CONCAT(JSON_PARSE('[10001,10002]'),JSON_PARSE('[10003,10004]'));

             array_concat
-----------------------------------
 [10001,10002,10003,10004]
(1 row)
```

Nell'esempio seguente viene utilizzata la manipolazione di array con la funzione SUBARRAY che restituisce un sottoinsieme dell'array di input.

```
SELECT SUBARRAY(ARRAY('a', 'b', 'c', 'd', 'e', 'f'), 2, 3);

   subarray
---------------
 ["c","d","e"]
(1 row))
```

Nell'esempio seguente vengono uniti più livelli di array in un singolo array tramite ARRAY_FLATTEN:

```
SELECT x, ARRAY_FLATTEN(x) FROM (SELECT ARRAY(1, ARRAY(2, ARRAY(3, ARRAY())))  AS x);

     x           | array_flatten
 ----------------+---------------
 [1,[2,[3,[]]]]  | [1,2,3]
(1 row)
```

Le funzioni di array ARRAY_CONCAT e ARRAY_FLATTEN utilizzano regole di digitazione dinamica. Restituiscono un null invece di un errore se l'input non è un array. La funzione GET_ARRAY_LENGTH restituisce la lunghezza di un array SUPER dato un percorso oggetto o array.

```
SELECT c_name
FROM customer_orders_lineitem
WHERE GET_ARRAY_LENGTH(c_orders) = (
    SELECT MAX(GET_ARRAY_LENGTH(c_orders))
    FROM customer_orders_lineitem
    );
```

Nell'esempio seguente una stringa viene suddivisa in un array di stringhe tramite SPLIT_TO_ARRAY. La funzione utilizza un delimitatore come parametro opzionale. Se non è assente alcun delimitatore, il valore di default è una virgola.

```
SELECT SPLIT_TO_ARRAY('12|345|6789', '|');

   split_to_array
---------------------
 ["12","345","6789"]
(1 row)
```

Configurazioni di SUPER

Notare le seguenti considerazioni relative alle configurazioni SUPER quando si utilizza il tipo di dati SUPER di Amazon Redshift e PartiQL.

Modalità permissiva e rigorosa per SUPER

Quando si esegue una query sui dati SUPER, l'espressione del percorso potrebbe non corrispondere alla struttura di dati SUPER effettiva. Se si prova ad accedere a un membro inesistente di un oggetto

o di un elemento di un array, Amazon Redshift restituisce un valore NULL se la query viene eseguita nella modalità permissiva di default. Se si esegue la query in modalità rigorosa, Amazon Redshift restituisce un errore. I seguenti parametri di sessione possono essere impostati per attivare o disattivare la modalità permissiva.

Nell'esempio seguente vengono utilizzati i parametri di sessione per abilitare la modalità permissiva.

```
SET navigate_super_null_on_error=ON;  --default lax mode for navigation

SET cast_super_null_on_error=ON;  --default lax mode for casting

SET parse_super_null_on_error=OFF;  --default strict mode for ingestion
```

Accesso ai campi JSON con nomi di campi o attributi in maiuscolo o maiuscolo e minuscolo

Quando i nomi degli attributi JSON sono in maiuscolo o in lettere maiuscole e minuscole, è necessario essere in grado di navigare nelle strutture di tipo SUPER facendo distinzione tra maiuscole e minuscole. Per farlo, puoi configurare `enable_case_sensitive_identifier` su TRUE e racchiudere i nomi degli attributi in maiuscolo e in lettere maiuscole e minuscole con virgolette doppie. Puoi anche configurare `enable_case_sensitive_super_attribute` su TRUE. In questo caso, puoi utilizzare nomi di attributi in maiuscolo e in maiuscolo e minuscolo nelle query senza racchiuderli tra virgolette doppie.

Nell'esempio seguente viene mostrato come impostare `enable_case_sensitive_identifier` per eseguire query sui dati.

```
SET enable_case_sensitive_identifier to TRUE;

-- Accessing JSON attribute names with uppercase and mixedcase names
SELECT json_table.data."ITEMS"."Name",
       json_table.data."price"
FROM
  (SELECT json_parse('{"ITEMS":{"Name":"TV"}, "price": 345}') AS data) AS json_table;

 Name | price
------+-------
 "TV" | 345
(1 row)
```

```
RESET enable_case_sensitive_identifier;

-- After resetting the above configuration, the following query accessing JSON
 attribute names with uppercase and mixedcase names should return null (if in lax
 mode).
SELECT json_table.data."ITEMS"."Name",
       json_table.data."price"
FROM
  (SELECT json_parse('{"ITEMS":{"Name":"TV"}, "price": 345}') AS data) AS json_table;

 name | price
------+-------
      | 345
(1 row)
```

Nell'esempio seguente viene mostrato come impostare `enable_case_sensitive_super_attribute` per eseguire query sui dati.

```
SET enable_case_sensitive_super_attribute to TRUE;
 -- Accessing JSON attribute names with uppercase and mixedcase names

SELECT json_table.data.ITEMS.Name,
       json_table.data.price
FROM
  (SELECT json_parse('{"ITEMS":{"Name":"TV"}, "price": 345}') AS data) AS json_table;

 name | price
 -----+-------
 "TV" | 345
(1 row)

 RESET enable_case_sensitive_super_attribute;

 -- After resetting enable_case_sensitive_super_attribute, the query now returns NULL
 for ITEMS.Name (if in lax mode).

SELECT json_table.data.ITEMS.Name,
       json_table.data.price
FROM
  (SELECT json_parse('{"ITEMS":{"Name":"TV"}, "price": 345}') AS data) AS json_table;

 name | price
 -----+-------
```

```
      | 345
(1 row)
```

Opzioni di analisi per SUPER

Quando si utilizza la funzione JSON_PARSE per analizzare le stringhe JSON in valori SUPER, si applicano alcune restrizioni:

- Lo stesso nome di attributo non può essere visualizzato nello stesso oggetto, ma può essere visualizzato in un oggetto annidato. L'opzione di configurazione `json_parse_dedup_attributes` consente a JSON_PARSE di mantenere solo l'ultima occorrenza di attributi duplicati anziché restituire un errore.
- I valori di stringa non possono superare la dimensione massima varchar del sistema di 65535 byte. L'opzione di configurazione `json_parse_truncate_strings` consente a JSON_PARSE() di troncare automaticamente le stringhe più lunghe di questo limite senza restituire un errore. Questo comportamento influisce solo sui valori di stringa e non sui nomi degli attributi.

Per ulteriori informazioni sull'utilizzo della funzione JSON_PARSE, consulta Funzione JSON_PARSE.

Gli esempi seguenti mostrano come configurare l'opzione di configurazione `json_parse_dedup_attributes` per il comportamento di default di restituire un errore per gli attributi duplicati.

```
SET json_parse_dedup_attributes=OFF;  --default behavior of returning error instead of
 de-duplicating attributes
```

Gli esempi seguenti mostrano come impostare l'opzione di configurazione `json_parse_truncate_strings` per il comportamento di default di restituire un errore per le stringhe più lunghe di questo limite.

```
SET json_parse_truncate_strings=OFF;  --default behavior of returning error instead of
 truncating strings
```

Limitazioni

Quando si utilizza il tipo di dati SUPER, è importante tenere presenti le seguenti limitazioni:

- Non è possibile definire colonne SUPER come chiave di distribuzione o di ordinamento.

- Un singolo oggetto SUPER può contenere fino a 16 MB di dati.
- Un valore individuale all'interno di un oggetto SUPER è limitato alla lunghezza massima del tipo Amazon Redshift corrispondente. Ad esempio, un valore di stringa singola caricato su SUPER è limitato alla lunghezza massima di VARCHAR di 65535 byte.
- Non è possibile eseguire operazioni di aggiornamento parziale o trasformazione sulle colonne SUPER.
- Non è possibile utilizzare il tipo di dati SUPER e il relativo alias nei join di destra o nei join esterni completi.
- Il tipo di dati SUPER non supporta XML come formato di serializzazione in ingresso o in uscita.
- Nella clausola FROM di una query secondaria (correlata o meno) che fa riferimento a una variabile di tabella per l'annullamento della nidificazione, la query può fare riferimento solo alla tabella padre e non ad altre tabelle.
- Limitazioni di casting

  È possibile eseguire il casting di valori SUPER da e verso altri tipi di dati con alcune eccezioni.

  - Amazon Redshift non fa alcuna differenza tra numeri interi e decimali di scala 0.
  - Se la scala non è zero, il tipo di dati SUPER ha lo stesso comportamento degli altri tipi di dati Amazon Redshift, ad eccezione del fatto che Amazon Redshift converte gli errori correlati a SUPER in null, come mostrato nell'esempio seguente.

```
SELECT 5::bool;
 bool
-------
 True
(1 row)

SELECT 5::decimal::bool;
ERROR:  cannot cast type numeric to boolean

SELECT 5::super::bool;
 bool
-------
 True
(1 row)

SELECT 5.0::bool;
ERROR:  cannot cast type numeric to boolean
```

```
SELECT 5.0::super::bool;
 bool
-------
(1 row)
```

- Amazon Redshift non esegue il casting dei tipi di data e ora sul tipo di dati SUPER. Amazon Redshift può eseguire il casting dei tipi di dati di data e ora solo dal tipo di dati SUPER, come mostrato nell'esempio seguente.

```
SELECT o.o_orderdate FROM customer_orders_lineitem c,c.c_orders o;
  order_date
------------------
 "2001-09-08"
(1 row)


SELECT JSON_TYPEOF(o.o_orderdate) FROM customer_orders_lineitem c,c.c_orders o;
 json_typeof
-----------------
 string
(1 row)


SELECT o.o_orderdate::date FROM customer_orders_lineitem c,c.c_orders o;
 order_date
----------------
 2001-09-08
(1 row)


--date/time cannot be cast to super
SELECT '2019-09-09'::date::super;
ERROR:  cannot cast type date to super
```

- Il casting da valori non scalari (oggetto e array) a stringa restituisce NULL. Per serializzare correttamente questi valori non scalari, non eseguirne il casting. Utilizzare invece `json_serialize` per eseguire il casting di valori non scalari. La funzione `json_serialize` restituisce un varchar. In genere, non è necessario eseguire il casting di valori non scalari su varchar poiché Amazon Redshift serializza implicitamente come mostrato nel primo esempio seguente.

```
SELECT r_nations FROM region_nations WHERE r_regionkey=300;
```

```
   r_nations
----------------
 [1,"abc",null]
(1 row)

SELECT r_nations::varchar FROM region_nations WHERE r_regionkey=300;
 r_nations
-------------
(1 row)

SELECT JSON_SERIALIZE(r_nations) FROM region_nations WHERE r_regionkey=300;
 json_serialize
-----------------
 [1,"abc",null]
(1 row)
```

- Per i database senza distinzione tra maiuscole e minuscole, Amazon Redshift non supporta il tipo di dati SUPER. Per le colonne senza distinzione tra maiuscole e minuscole, Amazon Redshift non esegue il casting al tipo SUPER. Pertanto, Amazon Redshift non supporta le colonne SUPER che interagiscono con colonne senza distinzione tra maiuscole e minuscole che attivano il casting.

- Amazon Redshift non supporta funzioni volatili, come RANDOM ( ) o TIMEOFDAY ( ), nelle query secondarie che annullano la nidificazione di una tabella esterna o un lato sinistro (LHS) delle funzioni IN con tali query secondarie.

Utilizzo del tipo di dati SUPER con viste materializzate

Amazon Redshift estende la capacità delle viste materializzate per lavorare con il tipo di dati SUPER e PartiQL nelle viste materializzate. Le query SQL e PartiQL possono essere precalcolate utilizzando viste materializzate incrementali. Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

Dopo aver archiviato i dati senza schema e semistrutturati in SUPER, è possibile utilizzare le viste materializzate PartiQL per ispezionare i dati e suddividerli in viste materializzate.

Accelerazione delle query PartiQL

È possibile utilizzare le viste materializzate per accelerare le query PartiQL che spostano e/o annullano la nidificazione di dati gerarchici nelle colonne SUPER. Creare una o più viste materializzate per suddividere i valori SUPER in più colonne e utilizzare l'organizzazione a

colonne delle query analitiche di Amazon Redshift. Di conseguenza, le query fanno uso delle viste materializzate.

La vista materializzata essenzialmente estrae e normalizza i dati nidificati. Il livello di normalizzazione dipende da quanto sforzo si fa per trasformare i dati SUPER in dati colonnari convenzionali.

Divisione in colonne SUPER con viste materializzate

Nell'esempio seguente viene illustrata una vista materializzata che suddivide i dati nidificati con le colonne risultanti ancora del tipo di dati SUPER.

```
SELECT c.c_name, o.o_orderstatus
FROM customer_orders_lineitem c, c.c_orders o;
```

L'esempio seguente mostra una vista materializzata che crea colonne scalari Amazon Redshift convenzionali dai dati suddivisi.

```
SELECT c.c_name, c.c_orders[0].o_totalprice
FROM customer_orders_lineitem c;
```

È possibile creare una singola vista materializzata super_mv per accelerare entrambe le query.

Per rispondere alla prima query, è necessario materializzare l'attributo o_orderstatus. È possibile omettere l'attributo c_name perché non comporta la navigazione nidificata né l'annullamento della nidificazione. È inoltre necessario includere nella vista materializzata l'attributo c_custkey di customer_orders_lineitem per poter unire la tabella di base alla vista materializzata.

Per rispondere alla seconda query, è necessario materializzare anche l'attributo o_totalprice e l'indice di array o_idx di c_orders. Pertanto, è possibile accedere all'indice 0 di c_orders.

```
CREATE MATERIALIZED VIEW super_mv distkey(c_custkey) sortkey(c_custkey) AS (
  SELECT c_custkey, o.o_orderstatus, o.o_totalprice, o_idx
  FROM customer_orders_lineitem c, c.c_orders o AT o_idx
);
```

Gli attributi o_orderstatus e o_totalprice della vista materializzata super_mv sono SUPER.

La vista materializzata super_mv verrà aggiornata in modo incrementale dopo le modifiche apportate alla tabella di base customer_orders_lineitem.

```
REFRESH MATERIALIZED VIEW super_mv;
```

```
INFO: Materialized view super_mv was incrementally updated successfully.
```

Per riscrivere la prima query PartiQL come una normale query SQL, unire customer_orders_lineitem con super_mv come segue.

```
SELECT c.c_name, v.o_orderstatus
FROM customer_orders_lineitem c
JOIN super_mv v ON c.c_custkey = v.c_custkey;
```

Allo stesso modo, è possibile riscrivere la seconda query PartiQL. Nell'esempio seguente viene utilizzato un filtro su o_idx = 0.

```
SELECT c.c_name, v.o_totalprice
FROM customer_orders_lineitem c
JOIN super_mv v ON c.c_custkey = v.c_custkey
WHERE v.o_idx = 0;
```

Nel comando CREATE MATERIALIZED VIEW specificare c_custkey come chiave di distribuzione e chiave di ordinamento per super_mv. Amazon Redshift esegue un join di unione efficiente, supponendo che c_custkey sia anche la chiave di distribuzione e la chiave di ordinamento di customer_orders_lineitem. In caso contrario, è possibile specificare c_custkey come chiave di ordinamento e chiave di distribuzione di customer_orders_lineitem come segue.

```
ALTER TABLE customer_orders_lineitem
ALTER DISTKEY c_custkey, ALTER SORTKEY (c_custkey);
```

Utilizza l'istruzione EXPLAIN per verificare che Amazon Redshift esegua un join di unione nelle query riscritte.

```
EXPLAIN
      SELECT c.c_name, v.o_orderstatus
      FROM customer_orders_lineitem c JOIN super_mv v ON c.c_custkey = v.c_custkey;

      QUERY PLAN

 ------------------------------------------------------------------------------------------------
      XN Merge Join DS_DIST_NONE  (cost=0.00..34701.82 rows=1470776 width=27)
      Merge Cond: ("outer".c_custkey = "inner".c_custkey)
      ->  XN Seq Scan on mv_tbl__super_mv__0 derived_table2  (cost=0.00..14999.86
 rows=1499986 width=13)
```

```
      ->  XN Seq Scan on customer_orders_lineitem c  (cost=0.00..999.96 rows=99996
 width=30)
      (4 rows)
```

Creazione di colonne scalari Amazon Redshift da dati suddivisi

I dati senza schema archiviati in SUPER possono influire sulle prestazioni di Amazon Redshift. Ad esempio, filtrare i predicati o unire le condizioni in quanto le scansioni limitate all'intervallo non possono utilizzare in modo efficace le mappe di zona. Gli utenti e gli strumenti di BI possono utilizzare le viste materializzate come presentazione convenzionale dei dati e aumentare le prestazioni delle query analitiche.

La query seguente esegue la scansione della vista materializzata `super_mv` e filtra su `o_orderstatus`.

```
SELECT c.c_name, v.o_totalprice
FROM customer_orders_lineitem c
JOIN super_mv v ON c.c_custkey = v.c_custkey
WHERE v.o_orderstatus = 'F';
```

Ispezionare `stl_scan` per verificare che Amazon Redshift non sia in grado di utilizzare in modo efficace le mappe delle zone nella scansione con limiti di intervallo su `o_orderstatus`.

```
SELECT slice, is_rrscan FROM stl_scan
WHERE query = pg_last_query_id() AND perm_table_name LIKE '%super_mv%';

 slice | is_rrscan
-------+-----------
     0 | f
     1 | f
     5 | f
     4 | f
     2 | f
     3 | f
(6 rows)
```

Nell'esempio seguente viene eseguito l'adattamento della vista materializzata `super_mv` per creare colonne scalari fuori dai dati suddivisi. In questo caso, Amazon Redshift avvia `o_orderstatus` da SUPER a VARCHAR. Inoltre, specificare `o_orderstatus` come chiave di ordinamento per `super_mv`.

```
CREATE MATERIALIZED VIEW super_mv distkey(c_custkey) sortkey(c_custkey, o_orderstatus)
 AS (
  SELECT c_custkey, o.o_orderstatus::VARCHAR AS o_orderstatus, o.o_totalprice, o_idx
  FROM customer_orders_lineitem c, c.c_orders o AT o_idx
);
```

Dopo aver rieseguito la query, verificare che Amazon Redshift possa ora utilizzare le mappe delle zone.

```
SELECT v.o_totalprice
FROM super_mv v
WHERE v.o_orderstatus = 'F';
```

È possibile verificare che la scansione con limiti di intervallo ora utilizzi mappe di zona come indicato di seguito.

```
SELECT slice, is_rrscan FROM stl_scan
WHERE query = pg_last_query_id() AND perm_table_name LIKE '%super_mv%';

 slice | is_rrscan
-------+-----------
     0 | t
     1 | t
     2 | t
     3 | t
     4 | t
     5 | t
(6 rows)
```

Limitazioni per l'uso del tipo di dati SUPER con viste materializzate

Quando si utilizza il tipo di dati SUPER con viste materializzate per Amazon Redshift, si verificano le seguenti limitazioni.

Le viste materializzate in Amazon Redshift non hanno limitazioni specifiche rispetto a PartiQL o SUPER.

Per informazioni sulle limitazioni SQL generali durante la creazione di viste materializzate, consultare Limitazioni.

Per informazioni sulle limitazioni SQL generali durante l'aggiornamento incrementale delle viste materializzate , consultare Limitazioni per l'aggiornamento incrementale.

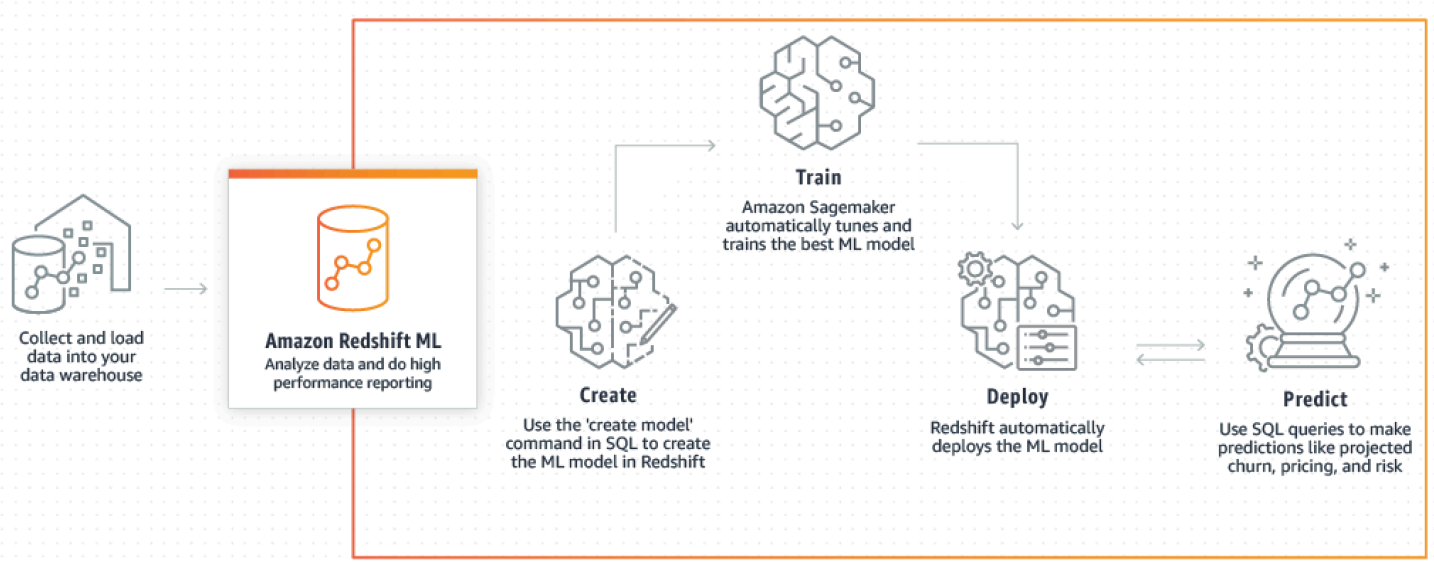
Utilizzo del machine learning in Amazon Redshift

Amazon Redshift Machine Learning (Amazon Redshift ML) è un efficace servizio basato su cloud che consente ad analisti e data scientist di qualunque livello di utilizzare la tecnologia di machine learning in tutta semplicità. I dati per cui si desidera addestrare un modello e i metadati associati agli input di dati vengono forniti ad Amazon Redshift. Quindi Amazon Redshift ML crea modelli che acquisiscono i pattern nei dati di input. È possibile quindi utilizzare questi modelli per generare previsioni per i nuovi dati di input senza dover sostenere costi aggiuntivi.

Come funziona Amazon Redshift ML con Amazon SageMaker

Amazon Redshift collabora con Amazon SageMaker Autopilot per ottenere automaticamente il modello migliore e rendere disponibile la funzione di previsione in Amazon Redshift.

Il seguente diagramma illustra come funziona Amazon Redshift ML.



Di seguito è riportato il flusso di lavoro generale:

1. Amazon Redshift esporta i dati di addestramento in Amazon S3.
2. Amazon SageMaker Autopilot preelabora i dati di addestramento. La pre-elaborazione esegue funzioni importanti, ad esempio l'imputazione di valori mancanti. Riconosce che alcune colonne sono di categoria (come il codice postale), le formatta correttamente per l'addestramento e svolge numerose altre attività. La scelta dei migliori preprocessori da applicare al set di dati di addestramento è di per sé un problema e Amazon SageMaker Autopilot automatizza la sua soluzione.

3. Amazon SageMaker Autopilot trova l'algoritmo e gli iperparametri dell'algoritmo che forniscono al modello le previsioni più accurate.
4. Amazon Redshift registra la funzione di previsione come funzione SQL nel cluster Amazon Redshift.
5. Quando esegui istruzioni CREATE MODEL, Amazon Redshift utilizza Amazon SageMaker per la formazione. Pertanto, vi è un costo associato per l'addestramento del modello. Si tratta di una voce distinta per Amazon SageMaker nella AWS fattura. Si paga anche lo spazio di archiviazione utilizzato in Amazon S3 per archiviare i dati di addestramento. L'inferenza che utilizza modelli creati con CREATE MODEL che possono essere compilati ed eseguiti sul cluster Redshift non verrà addebitata. Non ci sono costi aggiuntivi di Amazon Redshift per l'utilizzo di Amazon Redshift ML.

Argomenti

Panoramica del machine learning

Grazie ad Amazon Redshift ML, è possibile addestrare modelli di machine learning utilizzando istruzioni SQL e richiamarli nelle query SQL per la previsione.

Per scoprire come utilizzare Amazon Redshift ML, è possibile guardare il video Amazon Redshift ML.

Per informazioni sui prerequisiti per impostare il cluster Redshift, le autorizzazioni e la proprietà per l'utilizzo di Amazon Redshift ML, consultare le sezioni seguenti. Queste sezioni descrivono anche il funzionamento semplice dell'addestramento e delle previsioni in Amazon Redshift ML.

Come il machine learning può risolvere i problemi

Un modello di machine learning genera previsioni trovando modelli nei dati di addestramento e applicando questi modelli ai nuovi dati. Nel machine learning, è possibile addestrare questi modelli apprendendo i modelli che meglio spiegano i dati. Quindi i modelli vengono utilizzati per effettuare previsioni (dette anche inferenze) su nuovi dati. Il machine learning di solito è un processo iterativo

in cui è possibile continuare a migliorare l'accuratezza delle previsioni modificando iterativamente i parametri e migliorando i dati di addestramento. Se i dati cambiano, si verifica un altro addestramento dei nuovi modelli con il nuovo set di dati.

Per raggiungere i vari obiettivi aziendali, esistono diversi approcci fondamentali di machine learning.

Apprendimento supervisionato in Amazon Redshift ML

Amazon Redshift supporta l'apprendimento supervisionato, che è l'approccio più comune all'analisi aziendale avanzata. L'apprendimento supervisionato è l'approccio di machine learning preferito quando si dispone di un insieme consolidato di dati e di una comprensione di come i dati di input specifici prevedono vari risultati aziendali. Questi risultati sono talvolta chiamati etichette. In particolare, il set di dati è una tabella con attributi che comprendono funzioni (input) e destinazioni (output). Ad esempio, si supponga di disporre di una tabella che fornisce l'età e il codice postale per i clienti passati e attuali. Si supponga di avere anche un campo "attivo" che è true per i clienti attuali e false per i clienti che hanno sospeso la loro iscrizione. L'obiettivo del machine learning supervisionato è quello di individuare i modelli di età e codice postale che portano alla cessazione del cliente come rappresentato dai clienti le cui destinazioni sono "False". È possibile utilizzare questo modello per prevedere i clienti con probabilità di abbandono, ad esempio sospendendo la propria appartenenza, e potenzialmente offrire incentivi a restare.

Amazon Redshift supporta l'apprendimento supervisionato che include regressione, classificazione binaria e classificazione multiclass. Regressione si riferisce al problema di prevedere valori continui, come la spesa totale dei clienti. Classificazione binaria si riferisce al problema di prevedere uno di due risultati, ad esempio prevedere se un cliente abbandona o meno. La classificazione multiclass si riferisce al problema di prevedere uno di molti risultati, ad esempio prevedere l'articolo a cui potrebbe essere interessato un cliente. Gli analisti di dati e i data scientist possono utilizzarli per eseguire attività di apprendimento supervisionate per affrontare problemi che vanno dalla previsione, alla personalizzazione o alla previsione di abbandono dei clienti. L'apprendimento supervisionato può essere utilizzato anche in problemi quali la previsione di quali vendite stanno per terminare, la previsione dei ricavi, il rilevamento delle frodi e la previsione del valore di vita dei clienti.

Apprendimento senza supervisione in Amazon Redshift ML

L'apprendimento non supervisionato utilizza algoritmi di machine learning per analizzare e raggruppare dati di formazione senza etichetta. Gli algoritmi scoprono schemi o raggruppamenti nascosti. L'obiettivo è quello di modellare la struttura o la distribuzione sottostante nei dati per saperne di più sui dati.

Amazon Redshift supporta l'algoritmo di clustering K-Means per risolvere un problema di apprendimento senza supervisione. Questo algoritmo risolve i problemi di clustering in cui si desidera individuare i raggruppamenti nei dati. L'algoritmo K-Means tenta di trovare raggruppamenti discreti all'interno dei dati. I dati non classificati vengono raggruppati e ripartiti in base alle somiglianze e alle differenze. Raggruppando, l'algoritmo K-Means determina iterativamente i migliori centroidi e assegna ogni membro al centroide più vicino. I membri più vicini allo stesso centroide appartengono allo stesso gruppo. I membri di un gruppo sono il più simili possibile agli altri membri dello stesso gruppo e il più possibile diversi dai membri di altri gruppi. Ad esempio, l'algoritmo di clustering K-Means può essere utilizzato per classificare le città colpite da una pandemia o classificare le città in base alla popolarità dei prodotti di consumo.

Quando si utilizza l'algoritmo K-Means, si specifica un input k che specifica il numero di cluster da trovare nei dati. L'output di questo algoritmo è un insieme di centroidi k. Ogni punto dati appartiene a uno dei cluster k più vicini ad esso. Ogni cluster è descritto dal suo centroide. Il centroide può essere considerato come la media multidimensionale del cluster. L'algoritmo K-Means confronta le distanze per vedere quanto sono diversi i cluster l'uno dall'altro. Una distanza maggiore indica generalmente una maggiore differenza tra i cluster.

La preelaborazione dei dati è importante per K-Means, poiché garantisce che le caratteristiche del modello rimangano sulla stessa scala e producano risultati affidabili. Amazon Redshift supporta alcuni preprocessori K-Means per l'istruzione CREATE MODEL, ad esempio StandardScaler, e. MinMax NumericPassthrough Se non desideri applicare alcuna preelaborazione per K-means, scegli NumericPassthrough esplicitamente come trasformatore. Per ulteriori informazioni sui parametri K-Means, consultare CREA MODELLO con i parametri K-MEANS.

Per ulteriori informazioni su come eseguire un training senza supervisione con il clustering K-Means, guarda il seguente video: Unsupervised training with K-Means clustering (Training senza supervisione con il clustering K-Means).

Termini e concetti per Amazon Redshift ML

I seguenti termini vengono utilizzati per descrivere alcuni concetti di Amazon Redshift ML.

- Il machine learning in Amazon Redshift addestra un modello con un comando SQL. Amazon Redshift ML e Amazon SageMaker gestiscono tutte le conversioni dei dati, le autorizzazioni, l'utilizzo delle risorse e l'individuazione del modello corretto.
- L'addestramento è la fase in cui Amazon Redshift crea un modello di machine learning eseguendo un sottoinsieme specificato di dati nel modello. Amazon Redshift avvia automaticamente un processo di formazione in Amazon SageMaker e genera un modello.

- La previsione (chiamata anche inferenza) è l'uso del modello nelle query SQL di Amazon Redshift per prevedere i risultati. Al momento dell'inferenza, Amazon Redshift utilizza una funzione di previsione basata su modello come parte di una query più ampia per produrre previsioni. Le previsioni vengono calcolate localmente, nel cluster Redshift, fornendo così una velocità effettiva elevata, una bassa latenza e un costo zero.
- Con Bring your own model (BYOM), puoi utilizzare un modello addestrato all'esterno di Amazon Redshift con Amazon per l'inferenza nel database a livello locale in SageMaker Amazon Redshift. Amazon Redshift ML supporta l'utilizzo di BYOM in inferenza locale.
- L'inferenza locale viene utilizzata quando i modelli sono preaddestrati in Amazon SageMaker, compilati da Amazon SageMaker Neo e localizzati in Amazon Redshift ML. Per importare modelli supportati per l'inferenza locale in Amazon Redshift, utilizzare il comando CREATE MODEL. Amazon Redshift importa i SageMaker modelli preaddestrati chiamando Amazon Neo. SageMaker Compilare il modello e importarlo in Amazon Redshift. Utilizzare l'inferenza locale per accelerare la velocità e ridurre i costi.
- L'inferenza remota viene utilizzata quando Amazon Redshift richiama un endpoint modello distribuito in. SageMaker L'inferenza remota offre la flessibilità necessaria per richiamare tutti i tipi di modelli personalizzati e modelli di deep learning, come TensorFlow i modelli che hai creato e distribuito in Amazon. SageMaker

Altrettanto importanti sono le seguenti:

- Amazon SageMaker è un servizio di machine learning completamente gestito. Con Amazon SageMaker, data scientist e sviluppatori possono facilmente creare, addestrare e distribuire modelli direttamente in un ambiente ospitato pronto per la produzione. Per informazioni su Amazon SageMaker, consulta What is Amazon SageMaker nella Amazon SageMaker Developer Guide.
- Amazon SageMaker Autopilot è un set di funzionalità che addestra e ottimizza automaticamente i migliori modelli di machine learning per la classificazione o la regressione, in base ai tuoi dati. Si mantiene il pieno controllo e visibilità. Amazon SageMaker Autopilot supporta i dati di input in formato tabulare. Amazon SageMaker Autopilot offre la pulizia e la preelaborazione automatiche dei dati, la selezione automatica degli algoritmi per la regressione lineare, la classificazione binaria e la classificazione multiclasse. Supporta inoltre l'ottimizzazione automatica degli iperparametri (HPO), l'addestramento distribuito, l'istanza automatica e la selezione delle dimensioni del cluster. Per informazioni su Amazon SageMaker Autopilot, consulta Automatizza lo sviluppo di modelli con Amazon SageMaker Autopilot nella Amazon Developer Guide. SageMaker

Machine learning per principianti ed esperti

Amazon Redshift ML consente di addestrare i modelli con un singolo comando SQL CREATE MODEL. Il comando CREATE MODEL crea un modello utilizzato da Amazon Redshift per generare previsioni basate su modelli con costrutti SQL familiari.

Amazon Redshift ML è particolarmente utile quando non si dispone di competenze in machine learning, strumenti, linguaggi, algoritmi e API. Con il ML di Amazon Redshift, non è necessario eseguire alcuno dei carichi pesanti indifferenziati necessari per l'integrazione con un servizio di machine learning esterno. Amazon Redshift consente di risparmiare tempo per formattare e spostare i dati, gestire i controlli delle autorizzazioni o creare integrazioni, flussi di lavoro e script personalizzati. È possibile utilizzare facilmente gli algoritmi di machine learning più diffusi e semplificare le esigenze di addestramento che richiedono frequenti iterazioni dall'addestramento fino alla previsione. Amazon Redshift rileva automaticamente l'algoritmo migliore e ottimizza il modello migliore per il problema. È possibile fare previsioni dall'interno del cluster Amazon Redshift senza la necessità di spostare i dati da Amazon Redshift né di interfacciarsi e pagare per un altro servizio.

Amazon Redshift ML supporta analisti di dati e data scientist nell'utilizzo del machine learning. Inoltre, consente agli esperti di machine learning di utilizzare le loro conoscenze per guidare l'istruzione CREATE MODEL per utilizzare solo gli aspetti specificati. In questo modo, è possibile accelerare il tempo necessario a CREATE MODEL per trovare il miglior candidato e/o migliorare la precisione del modello.

L'istruzione CREATE MODEL offre flessibilità nel modo in cui è possibile specificare i parametri per il processo di addestramento. Ciò consente sia agli utenti principianti che a quelli esperti di machine learning di scegliere i preprocessori, gli algoritmi, i tipi di problemi o gli iperparametri preferiti. Ad esempio, un utente interessato all'abbandono del cliente potrebbe specificare all'istruzione CREATE MODEL che il tipo di problema è una classificazione binaria che funziona bene per l'abbandono del cliente. Quindi l'istruzione CREATE MODEL restringe la ricerca del modello migliore in modelli di classificazione binaria. Anche con la scelta dell'utente del tipo di problema, ci sono ancora molte opzioni che possono essere utilizzate dall'istruzione CREATE MODEL. Ad esempio, CREATE MODEL rileva e applica le migliori trasformazioni di pre-elaborazione e individua le migliori impostazioni dell'iperparametro.

Amazon Redshift ML semplifica la formazione trovando automaticamente il modello migliore con Amazon SageMaker Autopilot. Dietro le quinte, Amazon SageMaker Autopilot addestra e ottimizza automaticamente il miglior modello di machine learning in base ai dati forniti. Amazon SageMaker Neo compila quindi il modello di formazione e lo rende disponibile per la previsione nel cluster

Redshift. Quando si esegue una query di inferenza di machine learning utilizzando un modello addestrato, la query la query può utilizzare le enormi capacità di elaborazione in parallelo di Amazon Redshift. Allo stesso tempo, la query può utilizzare la previsione basata sul machine learning.

- Come principiante di machine learning, con una conoscenza generale dei diversi aspetti del machine learning, ad esempio preprocessori, algoritmi e iperparametri, utilizzare l'istruzione CREATE MODEL solo per gli aspetti specificati. Quindi è possibile ridurre il tempo necessario a CREATE MODEL per trovare il miglior candidato o migliorare la precisione del modello. Inoltre, è possibile aumentare il valore aziendale delle previsioni introducendo ulteriori conoscenze sul dominio, ad esempio il tipo di problema o l'obiettivo. Ad esempio, in uno scenario di abbandono del cliente, se il risultato "il cliente non è attivo" è raro, l'obiettivo F1 è spesso preferito all'obiettivo Precisione. Poiché i modelli ad alta precisione potrebbero prevedere "il cliente è attivo" per tutto il tempo, ciò si traduce in un'elevata precisione ma poco valore aziendale. Per informazioni sugli obiettivi F1, consulta JobObjectiveAutoML nell' SageMaker Amazon API Reference.

  Per ulteriori informazioni sulle opzioni di base per l'istruzione CREATE MODEL, consultare CREATE MODEL semplice.
- Come operatore avanzato di machine learning, è possibile specificare il tipo di problema e i preprocessori per determinate funzionalità (ma non tutte). Quindi CREATE MODEL segue i suggerimenti sugli aspetti specificati. Allo stesso tempo, CREATE MODEL rileva automaticamente i migliori preprocessori per le funzioni restanti e i migliori iperparametri. Per ulteriori informazioni su come limitare uno o più aspetti della pipeline di addestramento, consultare CREATE MODEL con guida per l'utente.
- Come esperto di machine learning, è possibile assumere il pieno controllo dell'addestramento e dell'ottimizzazione degli iperparametri. Quindi l'istruzione CREATE MODEL non prova a scoprire i preprocessori, gli algoritmi e gli iperparametri ottimali perché sei tu a fare tutte le scelte. Per ulteriori informazioni su come utilizzare l'istruzione CREATE MODEL con AUTO OFF, consultare Modelli CREATE XGBoost con AUTO OFF.
- In qualità di ingegnere dei dati, puoi importare un modello XGBoost preaddestrato in Amazon SageMaker e importarlo in Amazon Redshift per l'inferenza locale. Con Bring your own model (BYOM), puoi utilizzare un modello addestrato all'esterno di Amazon Redshift con Amazon per l'inferenza nel database a livello locale in SageMaker Amazon Redshift. Amazon Redshift ML supporta l'utilizzo di BYOM sia nell'inferenza locale che in quella remota.

  Per ulteriori informazioni su come utilizzare l'istruzione CREATE MODEL per l'inferenza locale o remota, consultare Bring your own model (BYOM) - inferenza locale.

In qualità di utente Amazon Redshift ML, è possibile scegliere una delle seguenti opzioni per addestrare e distribuire il proprio modello.

- Tipi di problema, consultare CREATE MODEL con guida per l'utente.
- Obiettivi, consultare CREATE MODEL con guida per l'utente o Modelli CREATE XGBoost con AUTO OFF.
- Tipi di modello, consultare Modelli CREATE XGBoost con AUTO OFF.
- Preprocessori, consultare CREATE MODEL con guida per l'utente.
- Iperparametri, consultare Modelli CREATE XGBoost con AUTO OFF.
- Porta il tuo modello (BYOM), consultare Bring your own model (BYOM) - inferenza locale.

Costi per l'utilizzo di Amazon Redshift ML

Amazon Redshift ML utilizza le risorse cluster esistenti per la previsione in modo da evitare ulteriori addebiti Amazon Redshift. Non sono previsti costi aggiuntivi Amazon Redshift per la creazione o l'utilizzo di un modello. La previsione avviene localmente nel cluster Redshift, quindi non è necessario pagare extra a meno che non sia necessario ridimensionare il cluster. Amazon Redshift ML utilizza Amazon SageMaker per addestrare il tuo modello, il che comporta un costo aggiuntivo associato.

Le funzioni di previsione eseguite all'interno del cluster Amazon Redshift non comportano costi supplementari. L'istruzione CREATE MODEL utilizza Amazon SageMaker e comporta un costo aggiuntivo. Il costo aumenta con il numero di celle nei dati di addestramento. Il numero di celle è il prodotto del numero di record (nella query di addestramento o nelle ore di tabella) moltiplicato per il numero di colonne. Ad esempio, quando una query SELECT dell'istruzione CREATE MODEL crea 10.000 record e 5 colonne, il numero di celle create è 50.000.

In alcuni casi, i dati di addestramento prodotti dalla query SELECT di CREATE MODEL superano il limite MAX_CELLS fornito (o il valore predefinito di 1 milione se non è stato impostato un limite). In questi casi, CREATE MODEL sceglie casualmente circa MAX_CELLS (cioè il "numero di colonne" record dal set di dati di addestramento). CREATE MODEL quindi esegue la formazione utilizzando queste tuple scelte casualmente. Il campionamento casuale assicura che il set di dati di addestramento ridotto non abbia alcun errore. Pertanto, impostando MAX_CELLS, è possibile controllare i costi di addestramento.

Quando si utilizza l'istruzione CREATE MODEL, è possibile utilizzare le opzioni MAX_CELLS e MAX_RUNTIME per controllare i costi, il tempo e la precisione potenziale del modello.

MAX_RUNTIME specifica la quantità massima di tempo che la formazione può impiegare SageMaker quando viene utilizzata l'opzione AUTO ON o OFF. I processi di addestramento spesso vengono completati prima di MAX_RUNTIME, in base alle dimensioni del set di dati. Dopo aver addestrato un modello, Amazon Redshift esegue ulteriori lavori in background per compilare e installare i modelli nel cluster. Pertanto, il completamento di CREATE MODEL può richiedere più tempo di MAX_RUNTIME. Tuttavia, MAX_RUNTIME limita la quantità di calcolo e il tempo utilizzati per addestrare il modello. SageMaker È possibile controllare lo stato del modello in qualsiasi momento utilizzando SHOW MODEL.

Quando esegui CREATE MODEL con AUTO ON, Amazon Redshift ML utilizza SageMaker Autopilot per esplorare in modo automatico e intelligente diversi modelli (o candidati) per trovare quello migliore. MAX_RUNTIME limita la quantità di tempo e calcolo impiegato. Se MAX_RUNTIME è impostato su un valore troppo basso, potrebbe non esserci abbastanza tempo per esplorare neanche un candidato. Se viene visualizzato l'errore "Il candidato pilota automatico non ha modelli", eseguire di nuovo CREATE MODEL con un valore maggiore di MAX_RUNTIME. Per ulteriori informazioni su questo parametro, consulta MaxAutoML JobRuntimeInSeconds nell'Amazon SageMaker API Reference.

Quando esegui CREATE MODEL con AUTO OFF, MAX_RUNTIME corrisponde a un limite per la durata dell'esecuzione del processo di formazione. SageMaker I processi di addestramento vengono spesso completati prima, a seconda delle dimensioni del set di dati e di altri parametri utilizzati, ad esempio num_rounds in MODEL_TYPE XGBOOST.

È inoltre possibile controllare i costi o ridurre i tempi di addestramento specificando un valore MAX_CELLS più piccolo quando si esegue CREATE MODEL. Una cella è una voce nel database. Ogni riga corrisponde a tutte le celle quante sono le colonne, che possono essere di larghezza fissa o variabile. MAX_CELLS limita il numero di celle e quindi il numero di esempi di addestramento utilizzati per addestrare il modello. Per impostazione predefinita, MAX_CELLS è impostato su 1 milione di celle. La riduzione di MAX_CELLS riduce il numero di righe dal risultato della query SELECT in CREATE MODEL che Amazon Redshift esporta e invia SageMaker per addestrare un modello. La riduzione di MAX_CELLS riduce così la dimensione del set di dati utilizzato per addestrare i modelli sia con AUTO ON che con AUTO OFF. Questo approccio aiuta a ridurre i costi e i tempi di addestramento dei modelli. Per visualizzare informazioni sulla formazione e sui tempi di fatturazione di uno specifico lavoro di formazione, scegli Lavori di formazione in Amazon SageMaker.

L'aumento di MAX_RUNTIME e MAX_CELLS spesso migliora la qualità del modello consentendo SageMaker di esplorare più candidati. In questo modo, SageMaker può richiedere più tempo per addestrare ciascun candidato e utilizzare più dati per addestrare modelli migliori. Se si

desidera un'iterazione o un'esplorazione del set di dati più rapida, utilizzare valori più bassi per MAX_RUNTIME e MAX_CELLS. Se si desidera una maggiore precisione dei modelli, utilizzare valori maggiori per MAX_RUNTIME e MAX_CELLS.

Per ulteriori informazioni sui costi associati a vari numeri di cellulare e i dettagli della versione di prova gratuita, consultare Prezzi di Amazon Redshift.

Nozioni di base su Amazon Redshift ML

Amazon Redshift ML rende più semplice agli utenti SQL la creazione, l'addestramento e l'implementazione dei modelli di machine learning grazie all'uso di comandi SQL familiari. Con Amazon Redshift ML, puoi usare i dati nel tuo cluster Redshift per addestrare modelli con Amazon. SageMaker Successivamente i modelli vengono localizzati ed è possibile fare le previsioni all'interno di un database Amazon Redshift. Attualmente Amazon Redshift ML supporta gli algoritmi di machine learning XGBoost (AUTO ON e OFF) e percezione multistrato (AUTO ON), K-Means (AUTO OFF) e Linear Learner.

Argomenti

- Configurare il cluster e le impostazioni di Amazon Redshift ML
- Utilizzo della spiegabilità del modello con Amazon Redshift ML
- Parametri di probabilità di Amazon Redshift ML
- Tutorial per Amazon Redshift ML

Configurare il cluster e le impostazioni di Amazon Redshift ML

Prima di lavorare con Amazon Redshift ML, completare la configurazione del cluster e configurare le autorizzazioni per l'utilizzo di Amazon Redshift ML.

Configurazione del cluster per l'utilizzo di Amazon Redshift ML

Prima di lavorare con Amazon Redshift ML, completare i prerequisiti seguenti.

In qualità di amministratore Amazon Redshift, effettuare le seguenti operazioni di configurazione una tantum:

Per eseguire una configurazione del cluster una tantum per Amazon Redshift ML

1. Crea un cluster Redshift utilizzando AWS Management Console o il AWS Command Line Interface ()AWS CLI. Assicurati di allegare la policy AWS Identity and Access Management (IAM) durante la creazione del cluster. Per ulteriori informazioni sulle autorizzazioni necessarie per utilizzare Amazon Redshift ML con SageMaker Amazon, consulta Autorizzazioni necessarie per utilizzare Amazon Redshift machine learning (ML) con Amazon. SageMaker
2. Crea il ruolo IAM richiesto per l'utilizzo di Amazon Redshift ML in uno dei modi seguenti:

   - Un metodo semplice consiste nel creare un ruolo IAM con le policy `AmazonS3FullAccess` e `AmazonSageMakerFullAccess` per l'uso con Amazon Redshift ML. Se intendi creare anche modelli di previsione, collega la policy `AmazonForecastFullAccess` anche al tuo ruolo.
   - È consigliabile creare un ruolo IAM tramite la console Amazon Redshift che dispone della policy `AmazonRedshiftAllCommandsFullAccess` con autorizzazioni per eseguire comandi SQL, ad esempio CREATE MODEL. Amazon Redshift utilizza un meccanismo semplice basato su API per creare in modo programmatico ruoli IAM per tuo conto. Account AWS Amazon Redshift associa automaticamente le policy AWS gestite esistenti al ruolo IAM. Questo approccio significa che puoi rimanere all'interno della console Amazon Redshift e non devi passare alla console IAM per la creazione di ruoli. Per ulteriori informazioni, consultare Creazione di un ruolo IAM come predefinito per Amazon Redshift.

     Quando viene creato un ruolo IAM come predefinito per il cluster, includere `redshift` come parte del nome della risorsa o utilizzare un tag specifico di RedShift per taggare tali risorse.

     Se il tuo cluster ha attivato il routing Amazon VPC avanzato, puoi utilizzare un ruolo IAM creato tramite la console Amazon Redshift. Questo ruolo IAM ha la policy `AmazonRedshiftAllCommandsFullAccess` allegata e aggiunge le seguenti autorizzazioni alla policy. Queste autorizzazioni aggiuntive consentono ad Amazon Redshift di creare ed eliminare un'elastic network interface (ENI) nel tuo account e di collegarla alle attività di compilazione in esecuzione su Amazon EC2 o Amazon ECS. In questo modo è possibile accedere agli oggetti presenti nei bucket Amazon S3 solo dall'interno di un cloud privato virtuale (VPC) con l'accesso a Internet bloccato.

```
{
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
```

```
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DeleteNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:CreateNetworkInterface",
            "ec2:ModifyNetworkInterfaceAttribute"
        ],
        "Resource": "*"
}
```

- Se si desidera creare un ruolo IAM con una policy più restrittiva, è possibile utilizzare la policy seguente. È inoltre possibile modificare questa policy per soddisfare le esigenze specifiche.

  Il bucket Amazon S3 `redshift-downloads/redshift-ml/` è la posizione in cui vengono memorizzati i dati di esempio utilizzati per altri passaggi ed esempi. È possibile rimuoverlo se non è necessario caricare dati da Amazon S3. In alternativa, sostituirlo con altri bucket Amazon S3 utilizzati per caricare i dati in Amazon Redshift.

  I valori *`your-account-id`*, *`your-role`* e *`your-s3-bucket`* sono quelli specificati dall'utente come parte del comando CREATE MODEL.

  (Facoltativo) Utilizza la sezione AWS KMS keys della policy di esempio se specifichi una AWS KMS chiave durante l'utilizzo di Amazon Redshift ML. Il valore *`your-kms-key`* è la chiave che si utilizza come parte del comando CREATE MODEL.

  ```
  {
      "Version": "2012-10-17",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "cloudwatch:PutMetricData",
                  "ecr:BatchCheckLayerAvailability",
                  "ecr:BatchGetImage",
                  "ecr:GetAuthorizationToken",
                  "ecr:GetDownloadUrlForLayer",
                  "logs:CreateLogGroup",
                  "logs:CreateLogStream",
                  "logs:DescribeLogStreams",
                  "logs:PutLogEvents",
                  "sagemaker:*Job*"
  ```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole",
                "s3:AbortMultipartUpload",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:iam::<your-account-id>:role/<your-role>",
                "arn:aws:s3:::<your-s3-bucket>/*",
                "arn:aws:s3:::redshift-downloads/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<your-s3-bucket>",
                "arn:aws:s3:::redshift-downloads"
            ]
        }
        // Optional section needed if you use AWS KMS keys.
        ,{
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:Encrypt",
                "kms:GenerateDataKey*"
             ],
             "Resource": [
                "arn:aws:kms:<your-region>:<your-account-id>:key/<your-kms-key>"
             ]
        }
    ]
```

```
}
```

3. Per consentire ad Amazon Redshift e SageMaker assumere il ruolo di interagire con altri servizi, aggiungi la seguente policy di fiducia al ruolo IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "redshift.amazonaws.com",
          "sagemaker.amazonaws.com",
          "forecast.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

4. (Facoltativo) Crea un bucket Amazon S3 e una chiave. AWS KMS Questi possono essere utilizzati da Amazon Redshift per archiviare i dati di formazione inviati ad Amazon SageMaker e ricevere il modello addestrato da Amazon. SageMaker
5. (Facoltativo) Creare diverse combinazioni di ruoli IAM e bucket Amazon S3 per controllare l'accesso a diversi gruppi di utenti.
6. (Facoltativo) Quando attivi il routing VPC per il tuo cluster Redshift, crea un endpoint Amazon S3 e un SageMaker endpoint per il VPC in cui si trova il cluster Redshift. Ciò fa sì che il traffico possa essere eseguito attraverso il VPC tra i servizi durante CREATE MODEL. Per ulteriori informazioni sul routing VPC, consultare Routing VPC avanzato in Amazon Redshift.

   Per ulteriori informazioni sulle autorizzazioni necessarie per specificare un VPC privato per il tuo processo di ottimizzazione degli iperparametri, consulta Autorizzazioni necessarie per utilizzare Amazon Redshift ML con Amazon. SageMaker

Per informazioni su come utilizzare l'istruzione CREATE MODEL per iniziare a creare modelli per diversi casi d'uso, consultare CREATE MODEL.

Gestione delle autorizzazioni e della proprietà

Proprio come per altri oggetti di database, come tabelle o funzioni, Amazon Redshift associa la creazione e l'utilizzo di modelli ML per accedere ai meccanismi di controllo. Esistono autorizzazioni separate per la creazione di un modello che esegue funzioni di previsione.

Gli esempi seguenti utilizzano due gruppi di utenti, `retention_analyst_grp` (creatore del modello) e `marketing_analyst_grp` (utente del modello) che illustrano come Amazon Redshift gestisce il controllo degli accessi. L'analista di conservazione crea modelli di machine learning che altri utenti possono utilizzare tramite autorizzazioni acquisite.

Un utente con privilegi avanzati può concedere autorizzazioni USER o GROUP per creare modelli di machine learning utilizzando la seguente istruzione.

```
GRANT CREATE MODEL TO GROUP retention_analyst_grp;
```

Gli utenti o i gruppi con questa autorizzazione possono creare un modello in qualsiasi schema del cluster se un utente dispone dell'autorizzazione CREATE per SCHEMA. Il modello di machine learning fa parte della gerarchia dello schema in modo simile a tabelle, viste, procedure e funzioni definite dall'utente.

Supponendo che esista già uno schema `demo_ml`, concedere ai due gruppi di utenti l'autorizzazione per lo schema come indicato di seguito.

```
GRANT CREATE, USAGE ON SCHEMA demo_ml TO GROUP retention_analyst_grp;
```

```
GRANT USAGE ON SCHEMA demo_ml TO GROUP marketing_analyst_grp;
```

Per consentire ad altri utenti di utilizzare la funzione di inferenza di machine learning, concedere l'autorizzazione EXECUTE. Nell'esempio seguente viene utilizzata l'autorizzazione EXECUTE per concedere a marketing_analyst_grp GROUP l'autorizzazione per l'uso del modello.

```
GRANT EXECUTE ON MODEL demo_ml.customer_churn_auto_model TO GROUP
 marketing_analyst_grp;
```

Utilizzare l'istruzione REVOKE con CREATE MODEL ed EXECUTE per revocare tali autorizzazioni da utenti o gruppi. Per ulteriori informazioni sui comandi di controllo delle autorizzazioni, consultare GRANT e REVOKE.

Utilizzo della spiegabilità del modello con Amazon Redshift ML

Con la spiegabili del modello in Amazon Redshift ML, utilizzi i valori di importanza delle funzionalità per capire in che modo ciascun attributo nei dati di formazione contribuisce al risultato previsto.

La spiegabilità del modello aiuta a migliorare i modelli di Machine Learning (ML) grazie all'illustrazione delle previsioni effettuate dai modelli. La spiegabilità del modello aiuta a spiegare come questi modelli fanno previsioni utilizzando un approccio di attribuzione delle funzionalità.

Amazon Redshift ML incorpora la spiegabilità del modello per fornire funzionalità di spiegazione del modello agli utenti di Amazon Redshift ML. Per ulteriori informazioni sulla spiegabilità dei modelli, consulta What Is Fairness and Model Explainability for Machine Learning Predictions? nella Amazon SageMaker Developer Guide.

La spiegabilità del modello monitora anche le inferenze effettuate dai modelli in produzione per la deriva dell'attribuzione delle funzionalità. Fornisce inoltre strumenti per aiutarti a generare report di governance dei modelli che puoi utilizzare per informare i team di rischio e conformità e le autorità di regolamentazione esterne.

Quando si specifica l'opzione AUTO ON o AUTO OFF quando si utilizza l'istruzione CREATE MODEL, al termine del processo di addestramento del modello, SageMaker viene creato l'output esplicativo. È possibile utilizzare la funzione EXPLAIN_MODEL per interrogare il report di spiegabilità in formato JSON. Per ulteriori informazioni, consulta Funzioni di machine learning.

Parametri di probabilità di Amazon Redshift ML

Nei problemi di apprendimento supervisionato, le etichette delle classi sono il risultato di previsioni che utilizzano i dati di input. Ad esempio, se utilizzi un modello per prevedere se un cliente si iscriverà nuovamente a un servizio di streaming, le possibili etichette sono "probabile" e "improbabile". Redshift ML offre parametri di probabilità che assegnano a ciascuna etichetta un valore di probabilità. Questo aiuta a prendere decisioni migliori sulla base dei risultati previsti. In Amazon Redshift ML, i parametri di probabilità sono disponibili quando si creano modelli AUTO ON con un tipo di problema di classificazione binaria o multiclasse. Se si omette il parametro AUTO ON, Redshift ML presuppone che il modello debba essere impostato su AUTO ON.

Creazione del modello

Durante la creazione di un modello, Amazon Redshift rileva automaticamente il tipo di modello e il tipo di problema. Se si tratta di un problema di classificazione, Redshift crea automaticamente

una seconda funzione di inferenza che è possibile utilizzare per generare il valore di probabilità di ciascuna etichetta. Il nome di questa seconda funzione di inferenza è il nome della prima funzione di inferenza seguito dalla stringa `_probabilities`. Ad esempio, se la prima funzione di inferenza è stata denominata `customer_churn_predict`, allora il nome della seconda funzione di inferenza sarà `customer_churn_predict_probabilities`. Questa funzione può essere quindi utilizzata per eseguire query sulle probabilità di ciascuna etichetta.

```
CREATE MODEL customer_churn_model
FROM customer_activity
    PROBLEM_TYPE BINARY_CLASSIFICATION
TARGET churn
FUNCTION customer_churn_predict
IAM_ROLE {default}
AUTO ON
SETTINGS ( S3_BUCKET '<DOC-EXAMPLE-BUCKET>'
```

Come generare le probabilità

Una volta che la funzione di generazione delle probabilità è pronta, l'esecuzione del comando restituisce un tipo SUPER che contiene array delle probabilità restituite e le relative etichette associate. Ad esempio, il risultato `"probabilities" : [0.7, 0.3], "labels" : ["False.", "True."]` indica che l'etichetta False ha una probabilità di 0,7, mentre l'etichetta True ha una probabilità di 0,3.

```
SELECT customer_churn_predict_probabilities(Account_length, Area_code,
            VMail_message, Day_mins, Day_calls, Day_charge,Eve_mins, Eve_calls,
            Eve_charge, Night_mins, Night_calls, Night_charge,Intl_mins, Intl_calls,
            Intl_charge, Cust_serv_calls)
FROM customer_activity;

customer_churn_predict_probabilities
 --------------------
 {"probabilities" : [0.7, 0.3], "labels" : ["False.", "True."]}
 {"probabilities" : [0.8, 0.2], "labels" : ["False.", "True."]}
 {"probabilities" : [0.75, 0.25], "labels" : ["True.", "False"]}
```

Gli array delle probabilità e delle etichette sono sempre mostrati in base al loro valore di probabilità in ordine decrescente. È possibile scrivere una query per ottenere solo l'etichetta prevista con la probabilità più alta separando i risultati SUPER restituiti dalla funzione di generazione delle probabilità.

```
SELECT prediction.labels[0], prediction.probabilities[0]
            FROM (SELECT customer_churn_predict_probabilities(Account_length,
 Area_code,
            VMail_message, Day_mins, Day_calls, Day_charge,Eve_mins, Eve_calls,
            Eve_charge, Night_mins, Night_calls, Night_charge,Intl_mins, Intl_calls,
            Intl_charge, Cust_serv_calls) AS prediction
FROM customer_activity);

  labels   | probabilities
-----------+--------------
 "False."  | 0.7
 "False."  | 0.8
 "True."   | 0.75
```

Per semplificare le query, è possibile memorizzare i risultati della funzione di previsione in una tabella.

```
CREATE TABLE churn_auto_predict_probabilities AS
            (SELECT customer_churn_predict_probabilities(Account_length, Area_code,
            VMail_message, Day_mins, Day_calls, Day_charge,Eve_mins, Eve_calls,
            Eve_charge, Night_mins, Night_calls, Night_charge,Intl_mins,
            Intl_calls, Intl_charge, Cust_serv_calls) AS prediction
FROM customer_activity);
```

È possibile eseguire query sulla tabella con i risultati per ottenere solo previsioni con una probabilità superiore a 0,7.

```
SELECT prediction.labels[0], prediction.probabilities[0]
FROM churn_auto_predict_probabilities
WHERE prediction.probabilities[0] > 0.7;

  labels   | probabilities
-----------+--------------
 "False."  | 0.8
 "True."   | 0.75
```

Utilizzando una notazione dell'indice è possibile ottenere la probabilità di un'etichetta specifica. L'esempio seguente restituisce le probabilità di tutte le etichette `True.`.

```
SELECT label, index, p.prediction.probabilities[index]
FROM churn_auto_predict_probabilities p, p.prediction.labels AS label AT index
```

```
WHERE label='True.';

  label  | index | probabilities
---------+-------+---------------
 "True." |     0 | 0.3
 "True." |     0 | 0.2
 "True." |     0 | 0.75
```

L'esempio seguente restituisce tutte le righe che includono un'etichetta `True` con una probabilità maggiore di 0,7, indicante il probabile abbandono del cliente.

```
SELECT prediction.labels[0], prediction.probabilities[0]
FROM churn_auto_predict_probabilities
WHERE prediction.probabilities[0] > 0.7 AND prediction.labels[0] = "True.";

labels     | probabilities
-----------+--------------
 "True."   | 0.75
```

Tutorial per Amazon Redshift ML

Con funzionalità di machine learning di Amazon Redshift, è possibile addestrare modelli di machine learning utilizzando istruzioni SQL e quindi richiamarli nelle query SQL per generare previsioni. Il machine learning in Amazon Redshift addestra un modello con un comando SQL. Amazon Redshift avvia automaticamente un processo di formazione in Amazon SageMaker e genera un modello. Dopo aver creato un modello, è possibile eseguire previsioni in Amazon Redshift utilizzando la funzione di previsione del modello.

Seguire la procedura descritta in questi tutorial per scoprire le funzionalità di funzionalità di machine learning di Amazon Redshift.

- Tutorial: Creazione di modelli di abbandono dei clienti
- Tutorial: Creazione di modelli di inferenza remota
- Tutorial: Creazione di modelli di clustering K-means
- Tutorial: Creazione di modelli di classificazione multi-classe
- Tutorial: Creazione di modelli XGBoost
- Tutorial: Creazione di modelli di regressione
- Tutorial: Creazione di modelli di regressione con Linear Learner

- Tutorial: Creazione di modelli di classificazione multi-classe con Linear Learner

Tutorial: Creazione di modelli di abbandono dei clienti

In questo tutorial, utilizzi Amazon Redshift ML per creare un modello di abbandono dei clienti con il comando CREATE MODEL ed eseguire query di previsione per gli scenari utente. Implementerai quindi le query utilizzando la funzione SQL generata dal comando CREATE MODEL.

È possibile usare un semplice comando CREATE MODEL per esportare i dati di addestramento, addestrare un modello, importare il modello e preparare una funzione di previsione Amazon Redshift. Utilizzare l'istruzione CREATE MODEL per specificare i dati di addestramento come tabella o istruzione SELECT.

L'esempio utilizza queste informazioni cronologiche per creare un modello di machine learning per le previsioni di abbandono dei clienti di un operatore di telefonia mobile. Innanzitutto, SageMaker addestra il tuo modello di apprendimento automatico e poi lo testa utilizzando le informazioni del profilo di un cliente arbitrario. Dopo la convalida del modello, Amazon SageMaker distribuisce il modello e la funzione di previsione su Amazon Redshift. È possibile utilizzare la funzione di previsione per prevedere se un cliente intende o meno abbandonare il servizio.

Esempi di casi d'uso

Puoi risolvere altri problemi di classificazione binaria utilizzando Amazon Redshift ML, ad esempio prevedere se un lead di vendita verrà chiuso o meno. Puoi anche prevedere se una transazione finanziaria è legale o meno.

Attività

- Prerequisiti
- Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift
- Passaggio 2: creazione del modello di machine learning
- Passaggio 3: esecuzione di previsioni con il modello

Prerequisiti

I prerequisiti per il completamento di questo sono elencati di seguito:

- Devi configurare un cluster Amazon Redshift per Amazon Redshift ML. A questo scopo, utilizza la documentazione per configurare il cluster e le impostazioni per l'amministrazione di Amazon Redshift ML.
- Il cluster Amazon Redshift usato per creare il modello e il bucket Amazon S3 usato per la gestione temporanea dei dati di addestramento e per l'archiviazione degli artefatti del modello devono trovarsi nella stessa regione AWS .
- Per scaricare i comandi SQL e i set di dati di esempio utilizzati nella documentazione, esegui una delle seguenti operazioni:
  - Scaricare i comandi SQL, il file delle attività dei clienti e il file Abalone.
  - Utilizzando il AWS CLI per Amazon S3, esegui il comando seguente. È possibile utilizzare il proprio percorso di destinazione.

```
aws s3 cp s3://redshift-downloads/redshift-ml/tutorial-scripts/redshift-ml-
tutorial.sql </target/path>
aws s3 cp s3://redshift-downloads/redshift-ml/customer_activity/
customer_activity.csv </target/path>
aws s3 cp s3://redshift-downloads/redshift-ml/abalone_xgb/abalone_xgb.csv </target/
path>
```

Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift

Utilizza l'editor di query v2 di Amazon Redshift per modificare ed eseguire query e visualizzare i risultati.

L'esecuzione delle seguenti query consente di creare una tabella denominata `customer_activity` e acquisisce il set di dati di esempio da Amazon S3.

```
DROP TABLE IF EXISTS customer_activity;

CREATE TABLE customer_activity (
state varchar(2),
account_length int,
area_code int,
phone varchar(8),
intl_plan varchar(3),
vMail_plan varchar(3),
vMail_message int,
day_mins float,
day_calls int,
```

```
day_charge float,
total_charge float,
eve_mins float,
eve_calls int,
eve_charge float,
night_mins float,
night_calls int,
night_charge float,
intl_mins float,
intl_calls int,
intl_charge float,
cust_serv_calls int,
churn varchar(6),
record_date date
);

COPY customer_activity
FROM 's3://redshift-downloads/redshift-ml/customer_activity/'
REGION 'us-east-1' IAM_ROLE default
FORMAT AS CSV IGNOREHEADER 1;
```

Passaggio 2: creazione del modello di machine learning

L'abbandono è l'input di destinazione in questo modello. Tutti gli altri input per il modello sono attributi che aiutano a creare una funzione per prevedere la perdita di clienti.

L'esempio seguente utilizza l'operazione CREATE MODEL per restituire un modello che prevede se un cliente sarà attivo, utilizzando input come l'età del cliente, il codice postale, le spese e i casi. Nell'esempio seguente, sostituisci *DOC-EXAMPLE-BUCKET* con il tuo bucket Amazon S3.

```
CREATE MODEL customer_churn_auto_model
FROM
    (
      SELECT state,
             account_length,
             area_code,
             total_charge/account_length AS average_daily_spend,
             cust_serv_calls/account_length AS average_daily_cases,
             churn
      FROM customer_activity
      WHERE  record_date < '2020-01-01'
     )
TARGET churn FUNCTION ml_fn_customer_churn_auto
```

```
IAM_ROLE default SETTINGS (
  S3_BUCKET '<DOC-EXAMPLE-BUCKET>'
);
```

La query SELECT nell'esempio precedente crea i dati di addestramento. La clausola TARGET specifica quale colonna è una "etichetta" di machine learning usato dall'operazione CREATE MODEL per imparare a eseguire le previsioni. La colonna di destinazione "abbandono" indica se il cliente ha ancora un'iscrizione attiva o se l'ha sospesa. Il campo S3_BUCKET è il nome del bucket Amazon S3 creato in precedenza. Il bucket Amazon S3 viene utilizzato per condividere dati e artefatti di formazione tra Amazon Redshift e Amazon. SageMaker Le colonne rimanenti sono le funzionalità (input) utilizzate per la previsione.

Per un riepilogo della sintassi e delle funzionalità di un caso d'uso di base del comando CREATE MODEL, consulta CREATE MODEL semplice.

Aggiunta delle autorizzazioni per la crittografia lato server (facoltativo)

Amazon Redshift per impostazione predefinita utilizza Amazon SageMaker Autopilot per la formazione. In particolare, Amazon Redshift esporta in modo sicuro i dati di addestramento nel bucket Amazon S3 specificato dal cliente. Se non si specifica `KMS_KEY_ID`, per impostazione predefinita i dati vengono crittografati utilizzando la crittografia lato server SSE-S3.

Quando crittografi l'input utilizzando la crittografia lato server con una chiave AWS KMS gestita (SSE-MMS), aggiungi le seguenti autorizzazioni:

```
{
    "Effect": "Allow",
    "Action": [
    "kms:Encrypt"
    "kms:Decrypt"
    ]
}
```

Per ulteriori informazioni sui SageMaker ruoli Amazon, consulta i SageMaker ruoli Amazon nella Amazon SageMaker Developer Guide.

Verifica dello stato dell'addestramento del modello (facoltativo)

È possibile utilizzare il comando SHOW MODEL per verificare quando il modello è pronto.

Utilizza l'operazione seguente per visualizzare lo stato del modello.

```
SHOW MODEL customer_churn_auto_model;
```

Di seguito è riportato un esempio di output dell'operazione precedente.

```
+--------------------------
+-----------------------------------------------------------------------------------------------------------
+
|           Key            |
                 Value
             |
+--------------------------
+-----------------------------------------------------------------------------------------------------------
+
|        Model Name        |
        customer_churn_auto_model
             |
|       Schema Name        |
                 public
             |
|          Owner           |
                 awsuser
             |
|      Creation Time       |
        Tue, 14.06.2022 17:15:52
             |
|       Model State        |
                TRAINING
             |
|                          |

             |
|      TRAINING DATA:      |

             |
|          Query           | SELECT STATE, ACCOUNT_LENGTH, AREA_CODE, TOTAL_CHARGE /
 ACCOUNT_LENGTH AS AVERAGE_DAILY_SPEND, CUST_SERV_CALLS / ACCOUNT_LENGTH AS
 AVERAGE_DAILY_CASES, CHURN |
|                          |
         FROM CUSTOMER_ACTIVITY
             |
|                          |
    WHERE RECORD_DATE < '2020-01-01'
             |
```

```
|      Target Column       |
                  CHURN
             |
|                          |

             |
|       PARAMETERS:        |

             |
|        Model Type        |
                  auto
             |
|       Problem Type       |

             |
|        Objective         |

             |
|     AutoML Job Name      |
     redshiftml-20220614171552640901
             |
|      Function Name       |
        ml_fn_customer_churn_auto
             |
|   Function Parameters    |                                                  state
 account_length area_code average_daily_spend average_daily_cases
                       |
| Function Parameter Types |
      varchar int4 int4 float8 int4
             |
|         IAM Role         |
          default-aws-iam-role
             |
|        S3 Bucket         |
             DOC-EXAMPLE-BUCKET
             |
|       Max Runtime        |
                  5400
             |
+--------------------------
+-----------------------------------------------------------------------------------------------------------
+
```

Al completamento dell'addestramento del modello, la variabile `model_state` diventa `Model is Ready` e la funzione di previsione diventa disponibile.

Passaggio 3: esecuzione di previsioni con il modello

È possibile utilizzare le istruzioni SQL per visualizzare le previsioni generate dal modello di previsione. In questo esempio, la funzione di previsione creata dall'operazione CREATE MODEL è denominata `ml_fn_customer_churn_auto`. Gli argomenti di input per la funzione di previsione corrispondono ai tipi di funzionalità, ad esempio varchar per `state` e integer per `account_length`. L'output della funzione di previsione è dello stesso tipo della colonna TARGET dell'istruzione CREATE MODEL.

1. Hai eseguito l'addestramento del modello usando dati anteriori alla data 01/01/2020, quindi ora usi la funzione di previsione sul set di test. La query seguente visualizza le previsioni relative alla possibilità di abbandono dei clienti registrati dopo la data 01/01/2020.

```
SELECT
    phone,
    ml_fn_customer_churn_auto(
        state,
        account_length,
        area_code,
        total_charge / account_length,
        cust_serv_calls / account_length
    ) AS active
FROM
    customer_activity
WHERE
    record_date > '2020-01-01';
```

2. L'esempio seguente utilizza la stessa funzione di previsione per un caso d'uso diverso. In questo caso, Amazon Redshift prevede la percentuale di abbandoni e non abbandoni tra clienti provenienti da stati diversi in cui la data di registrazione è successiva a 01/01/2020.

```
WITH predicted AS (
    SELECT
        state,
        ml_fn_customer_churn_auto(
            state,
            account_length,
            area_code,
            total_charge / account_length,
```

```
            cust_serv_calls / account_length
        ) :: varchar(6) AS active
    FROM
        customer_activity
    WHERE
        record_date > '2020-01-01'
)
SELECT
    state,
    SUM(
        CASE
            WHEN active = 'True.' THEN 1
            ELSE 0
        END
    ) AS churners,
    SUM(
        CASE
            WHEN active = 'False.' THEN 1
            ELSE 0
        END
    ) AS nonchurners,
    COUNT(*) AS total_per_state
FROM
    predicted
GROUP BY
    state
ORDER BY
    state;
```

3. Nell'esempio seguente viene utilizzata la funzione di previsione per il caso d'uso di previsione della percentuale di clienti persi in uno stato. In questo caso, Amazon Redshift prevede la percentuale di abbandono associata a una data di registrazione successiva a 01/01/2020.

```
WITH predicted AS (
    SELECT
        state,
        ml_fn_customer_churn_auto(
            state,
            account_length,
            area_code,
            total_charge / account_length,
            cust_serv_calls / account_length
        ) :: varchar(6) AS active
```

```
    FROM
        customer_activity
    WHERE
        record_date > '2020-01-01'
)
SELECT
    state,
    CAST((CAST((SUM(
        CASE
            WHEN active = 'True.' THEN 1
            ELSE 0
        END
    )) AS FLOAT) / CAST(COUNT(*) AS FLOAT)) AS DECIMAL (3, 2)) AS pct_churn,
    COUNT(*) AS total_customers_per_state
FROM
    predicted
GROUP BY
    state
ORDER BY
    3 DESC;
```

Argomenti correlati

Per ulteriori informazioni su Amazon Redshift ML, fare riferimento ai seguenti collegamenti:

- Costi per l'utilizzo di Amazon Redshift ML
- Comando CREATE MODEL
- Funzione EXPLAIN_MODEL

Per ulteriori informazioni sul machine learning, consulta la documentazione seguente:

- Panoramica del machine learning
- Machine learning per principianti ed esperti
- Che cos'è l'equità e la spiegabilità dei modelli per le previsioni di Machine Learning?

Tutorial: Creazione di modelli di inferenza remota

Il seguente tutorial illustra i passaggi per creare un modello Random Cut Forest precedentemente addestrato e distribuito in Amazon SageMaker, al di fuori di Amazon Redshift. L'algoritmo Random

Cut Forest rileva punti dati anomali all'interno di un set di dati. La creazione di un modello con inferenza remota ti consente di portare il tuo SageMaker modello Random Cut Forest in Amazon Redshift. Quindi, in Amazon Redshift, usi SQL per eseguire previsioni su un endpoint remoto. SageMaker

Puoi utilizzare un comando CREATE MODEL per importare un modello di machine learning da un SageMaker endpoint Amazon e preparare una funzione di previsione di Amazon Redshift. Quando utilizzi l'operazione CREATE MODEL, fornisci il nome dell'endpoint del modello di SageMaker machine learning.

In questo tutorial, crei un modello di machine learning di Amazon Redshift utilizzando un endpoint SageMaker modello. Una volta che il modello di machine learning è pronto, puoi utilizzarlo per eseguire previsioni in Amazon Redshift. Innanzitutto, si addestra e si crea un endpoint in Amazon SageMaker, quindi si ottiene il nome dell'endpoint. Puoi quindi utilizzare il comando CREATE MODEL per creare un modello con Amazon Redshift ML. Esegui infine previsioni sul modello utilizzando la funzione di previsione generata dal comando CREATE MODEL.

Esempi di casi d'uso

È possibile utilizzare modelli Random Cut Forest e l'inferenza remota per il rilevamento di anomalie in altri set di dati, ad esempio la previsione di un rapido aumento o diminuzione delle transazioni di e-commerce. È inoltre possibile prevedere variazioni significative relative al meteo l'attività sismica.

Attività

- Prerequisiti
- Fase 1: Implementazione del modello Amazon SageMaker
- Fase 2: Ottieni l'endpoint del SageMaker modello
- Passaggio 3: caricamento dei dati da Amazon S3 ad Amazon Redshift
- Passaggio 4: creazione di un modello con Amazon Redshift ML
- Passaggio 5: esecuzione di previsioni con il modello

Prerequisiti

I prerequisiti per il completamento di questo sono elencati di seguito:

- Hai completato la configurazione amministrativa di Amazon Redshift ML.

- Hai scaricato il set di dati dei taxi di New York, creato un bucket Amazon S3 e caricato i dati nel bucket Amazon S3.
- È necessario addestrare, implementare il SageMaker modello e l'endpoint e ottenere il nome dell'endpoint. SageMaker Utilizza questo AWS CloudFormation modello per effettuare automaticamente il provisioning di tutte le SageMaker risorse del tuo AWS account.

Fase 1: Implementazione del modello Amazon SageMaker

1. Per distribuire il modello, vai alla SageMaker console Amazon, scegli Istanze Notebook in Notebook nel pannello di navigazione.
2. Scegli Open Jupyter per il notebook Jupyter creato dal modello. CloudFormation
3. Scegli `bring-your-own-model-remote-inference.ipynb`.
4. Configura i parametri per archiviare l'input e l'output di addestramento in Amazon S3 sostituendo le righe seguenti con il bucket e il prefisso Amazon S3.

```
data_location=f"s3://{bucket}/{prefix}/",
output_path=f"s3://{bucket}/{prefix}/output",
```

5. Seleziona il pulsante di avanzamento rapido per eseguire tutte le celle.

Fase 2: Ottieni l'endpoint del modello SageMaker

Sulla SageMaker console Amazon, in Inference nel riquadro di navigazione, scegli Endpoints e trova il nome del tuo modello. Quando crei il modello di inferenza remota in Amazon Redshift, devi copiare il nome dell'endpoint del modello.

Passaggio 3: caricamento dei dati da Amazon S3 ad Amazon Redshift

Utilizza l'editor di query v2 di Amazon Redshift per eseguire i comandi SQL seguenti in Amazon Redshift. Questi comandi eliminano la tabella `rcf_taxi_data`, se esiste, creano una tabella con lo stesso nome e caricano il set di dati di esempio nella tabella.

```
DROP TABLE IF EXISTS public.rcf_taxi_data CASCADE;

CREATE TABLE public.rcf_taxi_data (ride_timestamp timestamp, nbr_passengers int);

COPY public.rcf_taxi_data
FROM
```

```
    's3://sagemaker-sample-files/datasets/tabular/anomaly_benchmark_taxi/
NAB_nyc_taxi.csv'
    IAM_ROLE default
    IGNOREHEADER 1
    FORMAT AS CSV;
```

Passaggio 4: creazione di un modello con Amazon Redshift ML

Esegui la seguente query per creare un modello in Amazon Redshift ML utilizzando l'endpoint del SageMaker modello ottenuto nel passaggio precedente. *randomcutforest-xxxxxxxxx*Sostituiscilo con il nome del tuo SageMaker endpoint.

```
CREATE MODEL public.remote_random_cut_forest
FUNCTION remote_fn_rcf(int)
RETURNS decimal(10, 6) SAGEMAKER '<randomcutforest-xxxxxxxxx>' IAM_ROLE default;
```

Controllo dello stato del modello (facoltativo)

È possibile utilizzare il comando SHOW MODEL per verificare quando il modello è pronto.

Per verificare lo stato del modello, utilizza la seguente operazione SHOW MODEL.

```
SHOW MODEL public.remote_random_cut_forest
```

L'output mostra il nome dell' SageMaker endpoint e della funzione.

```
+-------------------------+----------------------------------------+
|        Model Name       |        remote_random_cut_forest        |
+-------------------------+----------------------------------------+
|       Schema Name       |                public                  |
|          Owner          |                awsuser                 |
|      Creation Time      |       Wed, 15.06.2022 17:58:21         |
|       Model State       |                 READY                  |
|                         |                                        |
|       PARAMETERS:       |                                        |
|         Endpoint        |     <randomcutforest-xxxxxxxxx>        |
|      Function Name      |             remote_fn_rcf              |
|      Inference Type     |                Remote                  |
| Function Parameter Types |                int4                   |
|         IAM Role        |          default-aws-iam-role          |
+-------------------------+----------------------------------------+
```

Passaggio 5: esecuzione di previsioni con il modello

L'algoritmo Amazon SageMaker Random Cut Forest è progettato per rilevare punti dati anomali all'interno di un set di dati. In questo esempio, il modello è progettato per rilevare i picchi a livello di corse in taxi dovuti a eventi importanti. È possibile utilizzare il modello per prevedere eventi anomali generando un punteggio delle anomalie per ogni punto dati.

Utilizza la seguente query per calcolare i punteggi delle anomalie nell'intero set di dati relativo ai taxi. Nota: si fa riferimento alla funzione utilizzata nell'istruzione CREATE MODEL nel passaggio precedente.

```
SELECT
    ride_timestamp,
    nbr_passengers,
    public.remote_fn_rcf(nbr_passengers) AS score
FROM
    public.rcf_taxi_data;
```

Verifica della presenza di anomalie alte e basse (facoltativo)

Esegui la seguente query per trovare i punti dati con punteggi superiori a tre deviazioni standard rispetto al punteggio medio.

```
WITH score_cutoff AS (
    SELECT
        STDDEV(public.remote_fn_rcf(nbr_passengers)) AS std,
        AVG(public.remote_fn_rcf(nbr_passengers)) AS mean,
        (mean + 3 * std) AS score_cutoff_value
    FROM
        public.rcf_taxi_data
)
SELECT
    ride_timestamp,
    nbr_passengers,
    public.remote_fn_rcf(nbr_passengers) AS score
FROM
    public.rcf_taxi_data
WHERE
    score > (
        SELECT
            score_cutoff_value
        FROM
```

```
            score_cutoff
    )
ORDER BY
    2 DESC;
```

Esegui la seguente query per trovare i punti dati con punteggi superiori a tre deviazioni standard rispetto al punteggio medio.

```
WITH score_cutoff AS (
    SELECT
        STDDEV(public.remote_fn_rcf(nbr_passengers)) AS std,
        AVG(public.remote_fn_rcf(nbr_passengers)) AS mean,
        (mean - 3 * std) AS score_cutoff_value
    FROM
        public.rcf_taxi_data
)
SELECT
    ride_timestamp,
    nbr_passengers,
    public.remote_fn_rcf(nbr_passengers) AS score
FROM
    public.rcf_taxi_data
WHERE
    score < (
        SELECT
            score_cutoff_value
        FROM
            score_cutoff
    )
ORDER BY
    2 DESC;
```

Argomenti correlati

Per ulteriori informazioni su Amazon Redshift ML, fare riferimento ai seguenti collegamenti:

- Costi per l'utilizzo di Amazon Redshift ML
- Operazione CREATE MODEL
- Funzione EXPLAIN_MODEL

Per ulteriori informazioni sul machine learning, consulta la documentazione seguente:

- Panoramica del machine learning
- Machine learning per principianti ed esperti
- Che cos'è l'equità e la spiegabilità dei modelli per le previsioni di Machine Learning?

Tutorial: Creazione di modelli di clustering K-means

In questo tutorial, viene utilizzato Amazon Redshift ML per creare, addestrare e distribuire un modello di machine learning basato sull'algoritmo k-means. Questo algoritmo risolve i problemi di clustering in cui si desidera individuare i raggruppamenti nei dati. K-means aiuta a raggruppare i dati non ancora etichettati. Per ulteriori informazioni sul clustering K-means, consulta How K-means Clustering Works nella Amazon Developer Guide. SageMaker

Utilizzerai un'operazione CREATE MODEL per creare un modello K-means da un cluster Amazon Redshift. È possibile usare un comando CREATE MODEL per esportare i dati di addestramento, addestrare un modello, importare il modello e preparare una funzione di previsione Amazon Redshift. Usa l'istruzione CREATE MODEL per specificare i dati di addestramento come tabella o istruzione SELECT.

In questo tutorial, utilizzi K-means sul set di dati GDELT (Global Database of Events, Language, and Tone), che monitora le notizie a livello mondiale e i dati archiviati in ogni momento del giorno. K-means raggrupperà eventi con toni, soggetti o luoghi simili. I dati vengono archiviati in più file su Amazon Simple Storage Service, in due cartelle diverse. Le cartelle sono storiche, ovvero riferite agli 1979-2013, con aggiornamenti quotidiani riferiti agli anni 2013 e successivi. In questo esempio, viene usato formato storico e vengono acquisiti i dati relativi al 1979.

Esempi di casi d'uso

Puoi risolvere altri problemi di clustering con Amazon Redshift ML, ad esempio il raggruppamento di clienti con abitudini di visualizzazione simili su un servizio di streaming. Puoi anche utilizzare Redshift ML per prevedere il numero ottimale di centri di spedizione per un servizio di consegna.

Attività

- Prerequisiti
- Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift
- Passaggio 2: creazione del modello di machine learning
- Passaggio 3: esecuzione di previsioni con il modello

Prerequisiti

Per completare questo tutorial, è necessario completare la configurazione amministrativa di Amazon Redshift ML.

Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift

1. Utilizza l'editor di query v2 di Amazon Redshift per eseguire la seguente query. La query elimina la tabella `gdelt_data` nello schema pubblico, se esiste, e crea una tabella con lo stesso nome nello schema pubblico.

```
DROP TABLE IF EXISTS gdelt_data CASCADE;

CREATE TABLE gdelt_data (
    GlobalEventId bigint,
    SqlDate bigint,
    MonthYear bigint,
    Year bigint,
    FractionDate double precision,
    Actor1Code varchar(256),
    Actor1Name varchar(256),
    Actor1CountryCode varchar(256),
    Actor1KnownGroupCode varchar(256),
    Actor1EthnicCode varchar(256),
    Actor1Religion1Code varchar(256),
    Actor1Religion2Code varchar(256),
    Actor1Type1Code varchar(256),
    Actor1Type2Code varchar(256),
    Actor1Type3Code varchar(256),
    Actor2Code varchar(256),
    Actor2Name varchar(256),
    Actor2CountryCode varchar(256),
    Actor2KnownGroupCode varchar(256),
    Actor2EthnicCode varchar(256),
    Actor2Religion1Code varchar(256),
    Actor2Religion2Code varchar(256),
    Actor2Type1Code varchar(256),
    Actor2Type2Code varchar(256),
    Actor2Type3Code varchar(256),
    IsRootEvent bigint,
    EventCode bigint,
    EventBaseCode bigint,
    EventRootCode bigint,
```

```
    QuadClass bigint,
    GoldsteinScale double precision,
    NumMentions bigint,
    NumSources bigint,
    NumArticles bigint,
    AvgTone double precision,
    Actor1Geo_Type bigint,
    Actor1Geo_FullName varchar(256),
    Actor1Geo_CountryCode varchar(256),
    Actor1Geo_ADM1Code varchar(256),
    Actor1Geo_Lat double precision,
    Actor1Geo_Long double precision,
    Actor1Geo_FeatureID bigint,
    Actor2Geo_Type bigint,
    Actor2Geo_FullName varchar(256),
    Actor2Geo_CountryCode varchar(256),
    Actor2Geo_ADM1Code varchar(256),
    Actor2Geo_Lat double precision,
    Actor2Geo_Long double precision,
    Actor2Geo_FeatureID bigint,
    ActionGeo_Type bigint,
    ActionGeo_FullName varchar(256),
    ActionGeo_CountryCode varchar(256),
    ActionGeo_ADM1Code varchar(256),
    ActionGeo_Lat double precision,
    ActionGeo_Long double precision,
    ActionGeo_FeatureID bigint,
    DATEADDED bigint
);
```

2. La seguente query carica i dati di esempio nella tabella `gdelt_data`.

```
COPY gdelt_data
FROM 's3://gdelt-open-data/events/1979.csv'
REGION 'us-east-1'
IAM_ROLE default
CSV
DELIMITER '\t';
```

Verifica dei dati di addestramento (facoltativo)

Per vedere in base a quali dati verrà addestrato il modello, usa la seguente query.

```
SELECT
    AvgTone,
    EventCode,
    NumArticles,
    Actor1Geo_Lat,
    Actor1Geo_Long,
    Actor2Geo_Lat,
    Actor2Geo_Long
FROM
    gdelt_data LIMIT 100;
```

Passaggio 2: creazione del modello di machine learning

Nell'esempio seguente viene utilizzato il comando CREATE MODEL per creare un modello che raggruppa i dati in sette cluster. Il valore K è il numero di cluster in cui sono suddivisi i punti dati. Il modello classifica i punti dati in cluster in cui i punti dati sono più simili tra loro. Suddividendo i punti dati in gruppi, l'algoritmo K-Means determina iterativamente il centroide del cluster migliore. L'algoritmo assegna quindi ogni punto dati al centroide del cluster più vicino. I membri più vicini allo stesso centroide del cluster appartengono allo stesso gruppo. I membri di un gruppo sono il più simili possibile agli altri membri dello stesso gruppo e il più possibile diversi dai membri di altri gruppi. Il valore K è soggettivo e dipende dai metodi che misurano le somiglianze tra i punti dati. È possibile modificare il valore K per uniformare le dimensioni dei cluster se i cluster sono distribuiti in modo non uniforme.

Nell'esempio seguente, sostituisci *DOC-EXAMPLE-BUCKET* con il tuo bucket Amazon S3.

```
CREATE MODEL news_data_clusters
FROM
    (
        SELECT
            AvgTone,
            EventCode,
            NumArticles,
            Actor1Geo_Lat,
            Actor1Geo_Long,
            Actor2Geo_Lat,
            Actor2Geo_Long
        FROM
            gdelt_data
    ) FUNCTION news_monitoring_cluster
    IAM_ROLE default
```

```
    AUTO OFF
    MODEL_TYPE KMEANS
    PREPROCESSORS 'none'
    HYPERPARAMETERS DEFAULT
    EXCEPT
    (K '7')
    SETTINGS (S3_BUCKET '<DOC-EXAMPLE-BUCKET>');
```

Verifica dello stato dell'addestramento del modello (facoltativo)

È possibile utilizzare il comando SHOW MODEL per verificare quando il modello è pronto.

Per verificare lo stato del modello, usa la seguente operazione SHOW MODEL e verifica se `Model State` è `Ready`.

```
SHOW MODEL NEWS_DATA_CLUSTERS;
```

Se il modello è pronto, l'output dell'operazione precedente dovrebbe indicare che `Model State` è `Ready`. Di seguito è riportato un esempio di output dell'operazione SHOW MODEL.

```
+--------------------------
+-------------------------------------------------------------------------------------------------
+
|        Model Name        |
 news_data_clusters                                  |
+--------------------------
+-------------------------------------------------------------------------------------------------
+
|       Schema Name        |                                                 public
                                        |
|          Owner           |                                               awsuser
                                        |
|      Creation Time       |                                     Fri, 17.06.2022
 16:32:19                                    |
|       Model State        |                                                 READY
                                        |
|        train:msd         |                                          2973.822754
                                        |
|      train:progress      |                                           100.000000
                                        |
|     train:throughput     |                                         237114.875000
                                        |
```

```
|      Estimated Cost      |                                                    0.004983
                                        |
|                          |
                                        |
|      TRAINING DATA:      |
                                        |
|          Query           | SELECT AVGTONE, EVENTCODE, NUMARTICLES, ACTOR1GEO_LAT,
 ACTOR1GEO_LONG, ACTOR2GEO_LAT, ACTOR2GEO_LONG |
|                          |                                                FROM GDELT_DATA
                                        |
|                          |
                                        |
|       PARAMETERS:        |
                                        |
|        Model Type        |                                                     kmeans
                                        |
|    Training Job Name     |
 redshiftml-20220617163219978978-kmeans                                |
|      Function Name       |
 news_monitoring_cluster                                      |
|   Function Parameters    |       avgtone eventcode numarticles actor1geo_lat
 actor1geo_long actor2geo_lat actor2geo_long        |
| Function Parameter Types |                             float8 int8 int8 float8 float8
 float8 float8                          |
|         IAM Role         |                                        default-aws-iam-
role                                      |
|        S3 Bucket         |                                           DOC-EXAMPLE-
BUCKET                                     |
|       Max Runtime        |                                                5400
                                        |
|                          |
                                        |
|     HYPERPARAMETERS:     |
                                        |
|       feature_dim        |                                                 7
                                        |
|            k             |                                                 7
                                        |
+--------------------------
+-----------------------------------------------------------------------------------------------
+
```

Passaggio 3: esecuzione di previsioni con il modello

Identificazione dei cluster

È possibile trovare raggruppamenti discreti, noti anche come cluster, identificati nei dati dal modello. Un cluster è l'insieme di punti dati più vicini al centroide del cluster rispetto a qualsiasi altro centroide. Poiché il valore K rappresenta il numero di cluster nel modello, rappresenta anche il numero di centroidi del cluster. La seguente query identifica i cluster mostrando il cluster associato a ciascuno `globaleventid`.

```
SELECT
    globaleventid,
    news_monitoring_cluster (
        AvgTone,
        EventCode,
        NumArticles,
        Actor1Geo_Lat,
        Actor1Geo_Long,
        Actor2Geo_Lat,
        Actor2Geo_Long
    ) AS cluster
FROM
    gdelt_data;
```

Verifica della distribuzione dei dati

Puoi controllare la distribuzione dei dati tra i cluster per vedere se il valore K scelto ha causato una distribuzione uniforme dei dati. Utilizza la seguente query per determinare se i dati sono distribuiti uniformemente tra i cluster.

```
SELECT
    events_cluster,
    COUNT(*) AS nbr_events
FROM
    (
        SELECT
            globaleventid,
            news_monitoring_cluster(
                AvgTone,
                EventCode,
                NumArticles,
                Actor1Geo_Lat,
```

```
                Actor1Geo_Long,
                Actor2Geo_Lat,
                Actor2Geo_Long
            ) AS events_cluster
        FROM
            gdelt_data
    )
GROUP BY
    1;
```

Nota: è possibile modificare il valore K per uniformare le dimensioni dei cluster se i cluster sono distribuiti in modo non uniforme.

Determinazione dei centroidi dei cluster

Un punto dati è più vicino al relativo centroide del cluster rispetto a qualsiasi altro centroide. Pertanto, la ricerca dei centroidi consente di definire i cluster.

Esegui la seguente query per determinare i centroidi dei cluster in base al numero di articoli per codice evento.

```
SELECT
    news_monitoring_cluster (
        AvgTone,
        EventCode,
        NumArticles,
        Actor1Geo_Lat,
        Actor1Geo_Long,
        Actor2Geo_Lat,
        Actor2Geo_Long
    ) AS events_cluster,
    eventcode,
    SUM(numArticles) AS numArticles
FROM
    gdelt_data
GROUP BY
    1,
    2;
```

Visualizzazione delle informazioni sui punti dati in un cluster

Utilizza la seguente query per restituire i dati per i punti assegnati al quinto cluster. Gli articoli selezionati devono avere due soggetti.

```
SELECT
    news_monitoring_cluster (
        AvgTone,
        EventCode,
        NumArticles,
        Actor1Geo_Lat,
        Actor1Geo_Long,
        Actor2Geo_Lat,
        Actor2Geo_Long
    ) AS events_cluster,
    eventcode,
    actor1name,
    actor2name,
    SUM(numarticles) AS totalarticles
FROM
    gdelt_data
WHERE
    events_cluster = 5
    AND actor1name <> ' '
    AND actor2name <> ' '
GROUP BY
    1,
    2,
    3,
    4
ORDER BY
    5 desc;
```

Visualizzazione dei dati sugli eventi con soggetti aventi lo stesso codice etnico

La seguente query conteggia il numero di articoli scritti sugli eventi con un tono positivo. La query richiede inoltre che i due soggetti abbiano lo stesso codice etnico e restituisce il cluster a cui è assegnato ciascun evento.

```
SELECT
    news_monitoring_cluster (
        AvgTone,
        EventCode,
        NumArticles,
        Actor1Geo_Lat,
        Actor1Geo_Long,
        Actor2Geo_Lat,
        Actor2Geo_Long
```

```
    ) AS events_cluster,
    SUM(numarticles) AS total_articles,
    eventcode AS event_code,
    Actor1EthnicCode AS ethnic_code
FROM
    gdelt_data
WHERE
    Actor1EthnicCode = Actor2EthnicCode
    AND Actor1EthnicCode <> ' '
    AND Actor2EthnicCode <> ' '
    AND AvgTone > 0
GROUP BY
    1,
    3,
    4
HAVING
    (total_articles) > 4
ORDER BY
    1,
    2 ASC;
```

Argomenti correlati

Per ulteriori informazioni su Amazon Redshift ML, fare riferimento ai seguenti collegamenti:

- Costi per l'utilizzo di Amazon Redshift ML
- Operazione CREATE MODEL
- Funzione EXPLAIN_MODEL

Per ulteriori informazioni sul machine learning, consulta la documentazione seguente:

- Panoramica del machine learning
- Machine learning per principianti ed esperti
- Che cos'è l'equità e la spiegabilità dei modelli per le previsioni di Machine Learning?

Tutorial: Creazione di modelli di classificazione multi-classe

In questo tutorial, utilizzi Amazon Redshift ML per creare un modello di machine learning che risolva i problemi di classificazione multi-classe. L'algoritmo di classificazione multi-classe classifica i punti

dati in una delle tre o più classi. Implementerai quindi le query utilizzando la funzione SQL generata dal comando CREATE MODEL.

È possibile usare un comando CREATE MODEL per esportare i dati di addestramento, addestrare un modello, importare il modello e preparare una funzione di previsione Amazon Redshift. Usa l'istruzione CREATE MODEL per specificare i dati di addestramento come tabella o istruzione SELECT.

Per eseguire il tutorial, è possibile utilizzare il set di dati pubblico E-Commerce Sales Forecast, che include i dati di vendita di un rivenditore online del Regno Unito. Il modello generato fa riferimento ai clienti più attivi per uno speciale programma di fidelizzazione dei clienti. Con la classificazione multi-classe, è possibile utilizzare il modello per prevedere per quanti mesi un cliente sarà attivo in un periodo di 13 mesi. La funzione di previsione individua i clienti che si prevede siano attivi per 7 o più mesi ai fini dell'ammissione al programma.

Esempi di casi d'uso

Puoi risolvere altri problemi di classificazione multi-classe con Amazon Redshift ML, ad esempio la previsione relativa al prodotto più venduto di una linea di prodotti. Puoi anche prevedere quale frutto include un'immagine, ad esempio selezionando mele, pere o arance.

Attività

- Prerequisiti
- Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift
- Passaggio 2: creazione del modello di machine learning
- Passaggio 3: esecuzione di previsioni con il modello

Prerequisiti

Per completare questo tutorial, è necessario completare la configurazione amministrativa di Amazon Redshift ML.

Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift

Utilizza l'editor di query v2 di Amazon Redshift per eseguire le seguenti query. Queste query caricano i dati di esempio in Amazon Redshift.

1. La query seguente crea una tabella denominata `ecommerce_sales`.

```
CREATE TABLE IF NOT EXISTS ecommerce_sales (
    invoiceno VARCHAR(30),
    stockcode VARCHAR(30),
    description VARCHAR(60),
    quantity DOUBLE PRECISION,
    invoicedate VARCHAR(30),
    unitprice DOUBLE PRECISION,
    customerid BIGINT,
    country VARCHAR(25)
);
```

2. La seguente query copia i dati di esempio dal set di dati E-Commerce Sales Forecast nella tabella `ecommerce_sales`.

```
COPY ecommerce_sales
FROM
    's3://redshift-ml-multiclass/ecommerce_data.txt'
IAM_ROLE default
DELIMITER '\t'
IGNOREHEADER 1
REGION 'us-east-1'
MAXERROR 100;
```

Suddivisione dei dati

Quando crei un modello in Amazon Redshift ML, suddivide SageMaker automaticamente i dati in set di training e test, in SageMaker modo da determinare la precisione del modello. Suddividendo manualmente i dati in questa fase, sarai in grado di verificare l'accuratezza del modello assegnando un set di previsione aggiuntivo.

Utilizza l'istruzione SQL seguente per suddividere i dati in tre set per l'addestramento, la convalida e la previsione.

```
--creates table with all data
CREATE TABLE ecommerce_sales_data AS (
    SELECT
        t1.stockcode,
        t1.description,
        t1.invoicedate,
        t1.customerid,
```

```
        t1.country,
        t1.sales_amt,
        CAST(RANDOM() * 100 AS INT) AS data_group_id
    FROM
        (
            SELECT
                stockcode,
                description,
                invoicedate,
                customerid,
                country,
                SUM(quantity * unitprice) AS sales_amt
            FROM
                ecommerce_sales
            GROUP BY
                1,
                2,
                3,
                4,
                5
        ) t1
);

--creates training set
CREATE TABLE ecommerce_sales_training AS (
    SELECT
        a.customerid,
        a.country,
        a.stockcode,
        a.description,
        a.invoicedate,
        a.sales_amt,
        (b.nbr_months_active) AS nbr_months_active
    FROM
        ecommerce_sales_data a
        INNER JOIN (
            SELECT
                customerid,
                COUNT(
                    DISTINCT(
                        DATE_PART(y, CAST(invoicedate AS DATE)) || '-' || LPAD(
                            DATE_PART(mon, CAST(invoicedate AS DATE)),
                            2,
                            '00'
```

```
                        )
                    )
                ) AS nbr_months_active
            FROM
                ecommerce_sales_data
            GROUP BY
                1
        ) b ON a.customerid = b.customerid
    WHERE
        a.data_group_id < 80
);

--creates validation set
CREATE TABLE ecommerce_sales_validation AS (
    SELECT
        a.customerid,
        a.country,
        a.stockcode,
        a.description,
        a.invoicedate,
        a.sales_amt,
        (b.nbr_months_active) AS nbr_months_active
    FROM
        ecommerce_sales_data a
        INNER JOIN (
            SELECT
                customerid,
                COUNT(
                    DISTINCT(
                        DATE_PART(y, CAST(invoicedate AS DATE)) || '-' || LPAD(
                            DATE_PART(mon, CAST(invoicedate AS DATE)),
                            2,
                            '00'
                        )
                    )
                ) AS nbr_months_active
            FROM
                ecommerce_sales_data
            GROUP BY
                1
        ) b ON a.customerid = b.customerid
    WHERE
        a.data_group_id BETWEEN 80
        AND 90
```

```
);

--creates prediction set
CREATE TABLE ecommerce_sales_prediction AS (
    SELECT
        customerid,
        country,
        stockcode,
        description,
        invoicedate,
        sales_amt
    FROM
        ecommerce_sales_data
    WHERE
        data_group_id > 90);
```

Passaggio 2: creazione del modello di machine learning

In questo passaggio, si utilizza l'istruzione CREATE MODEL per creare il modello di machine learning utilizzando la classificazione multi-classe.

La seguente query crea il modello di classificazione multi-classe con il set di addestramento utilizzando l'operazione CREATE MODEL. Sostituisci *DOC-EXAMPLE-BUCKET* con il tuo bucket Amazon S3.

```
CREATE MODEL ecommerce_customer_activity
FROM
    (
        SELECT
            customerid,
            country,
            stockcode,
            description,
            invoicedate,
            sales_amt,
            nbr_months_active
        FROM
            ecommerce_sales_training
    ) TARGET nbr_months_active FUNCTION predict_customer_activity IAM_ROLE default
 PROBLEM_TYPE MULTICLASS_CLASSIFICATION SETTINGS (
        S3_BUCKET '<DOC-EXAMPLE-BUCKET>',
        S3_GARBAGE_COLLECT OFF
    );
```

In questa query, specifica il tipo di problema come `Multiclass_Classification`. L'obiettivo previsto per il modello è `nbr_months_active`. Al SageMaker termine dell'addestramento del modello, crea la funzione `predict_customer_activity` che utilizzerai per fare previsioni in Amazon Redshift.

Visualizzazione dello stato dell'addestramento del modello (facoltativo)

È possibile utilizzare il comando SHOW MODEL per verificare quando il modello è pronto.

Utilizza la seguente query per restituire i vari parametri del modello, inclusi lo stato e la precisione.

```
SHOW MODEL ecommerce_customer_activity;
```

Se il modello è pronto, l'output dell'operazione precedente dovrebbe indicare che `Model State` è `Ready`. Di seguito è riportato un esempio di output dell'operazione SHOW MODEL.

```
+--------------------------
+-------------------------------------------------------------------------------------------
+
|        Model Name        |
 ecommerce_customer_activity                          |
+--------------------------
+-------------------------------------------------------------------------------------------
+
|       Schema Name        |                                          public
                                  |
|          Owner           |                                          awsuser
                                  |
|      Creation Time       |                                Fri, 17.06.2022 19:02:15
                                  |
|       Model State        |                                            READY
                                  |
|   Training Job Status    |
 MaxAutoMLJobRuntimeReached                           |
|   validation:accuracy    |                                        0.991280
                                  |
|      Estimated Cost      |                                        7.897689
                                  |
|                          |
                                  |
|      TRAINING DATA:      |
                                  |
```

```
|          Query           | SELECT CUSTOMERID, COUNTRY, STOCKCODE, DESCRIPTION,
 INVOICEDATE, SALES_AMT, NBR_MONTHS_ACTIVE |
|                          |                                    FROM
 ECOMMERCE_SALES_TRAINING                                  |
|      Target Column       |                                         NBR_MONTHS_ACTIVE
                                    |
|                          |
                                    |
|       PARAMETERS:        |
                                    |
|        Model Type        |                                           xgboost
                                    |
|       Problem Type       |                                    MulticlassClassification
                                    |
|        Objective         |                                           Accuracy
                                    |
|     AutoML Job Name      |
 redshiftml-20220617190215268770                                    |
|      Function Name       |
 predict_customer_activity                                  |
|   Function Parameters    |               customerid country stockcode description
 invoicedate sales_amt               |
| Function Parameter Types |                           int8 varchar varchar varchar
 varchar float8                        |
|         IAM Role         |                                      default-aws-iam-role
                                    |
|        S3 Bucket         |                                          DOC-EXAMPLE-BUCKET
                                    |
|       Max Runtime        |                                             5400
                                    |
+--------------------------
+-------------------------------------------------------------------------------------------
+
```

Passaggio 3: esecuzione di previsioni con il modello

La seguente query mostra quali clienti si qualificano per il programma di fidelizzazione. Se il modello prevede che il cliente sia attivo per almeno sette mesi, tale modello seleziona il cliente per il programma fedeltà.

```
SELECT
    customerid,
    predict_customer_activity(
```

```
        customerid,
        country,
        stockcode,
        description,
        invoicedate,
        sales_amt
    ) AS predicted_months_active
FROM
    ecommerce_sales_prediction
WHERE
    predicted_months_active >= 7
GROUP BY
    1,
    2
LIMIT
    10;
```

Esecuzione di query di previsione sui dati di convalida (facoltativo)

Esegui le seguenti query di previsione sui dati di convalida per visualizzare il livello di precisione del modello.

```
SELECT
    CAST(SUM(t1.match) AS decimal(7, 2)) AS predicted_matches,
    CAST(SUM(t1.nonmatch) AS decimal(7, 2)) AS predicted_non_matches,
    CAST(SUM(t1.match + t1.nonmatch) AS decimal(7, 2)) AS total_predictions,
    predicted_matches / total_predictions AS pct_accuracy
FROM
    (
        SELECT
            customerid,
            country,
            stockcode,
            description,
            invoicedate,
            sales_amt,
            nbr_months_active,
            predict_customer_activity(
                customerid,
                country,
                stockcode,
                description,
                invoicedate,
```

```
                sales_amt
            ) AS predicted_months_active,
            CASE
                WHEN nbr_months_active = predicted_months_active THEN 1
                ELSE 0
            END AS match,
            CASE
                WHEN nbr_months_active <> predicted_months_active THEN 1
                ELSE 0
            END AS nonmatch
        FROM
            ecommerce_sales_validation
    )t1;
```

Previsione del numero di clienti che non riescono a entrare (opzionale)

La seguente query confronta il numero di clienti che si prevede siano attivi solo per 5 o 6 mesi. Il modello prevede che questi clienti perdano l'iscrizione al programma fedeltà. La query confronta quindi l'importo mancante all'iscrizione al programma con il numero che si prevede sia idoneo per il programma fedeltà. Questa query potrebbe essere utilizzata per formulare una decisione relativa all'abbassamento della soglia per il programma fedeltà. Puoi anche determinare se esiste un numero significativo di clienti che si prevede non rientrino per poco nel programma. Potresti quindi incoraggiare questi clienti ad aumentare la loro attività per ottenere l''iscrizione al programma fedeltà.

```
SELECT
    predict_customer_activity(
        customerid,
        country,
        stockcode,
        description,
        invoicedate,
        sales_amt
    ) AS predicted_months_active,
    COUNT(customerid)
FROM
    ecommerce_sales_prediction
WHERE
    predicted_months_active BETWEEN 5 AND 6
GROUP BY
    1
ORDER BY
    1 ASC
```

```
LIMIT
    10)
UNION
(SELECT
      NULL AS predicted_months_active,
    COUNT (customerid)
FROM
    ecommerce_sales_prediction
WHERE
    predict_customer_activity(
        customerid,
        country,
        stockcode,
        description,
        invoicedate,
        sales_amt
    ) >=7);
```

Argomenti correlati

Per ulteriori informazioni su Amazon Redshift ML, fare riferimento ai seguenti collegamenti:

- Costi per l'utilizzo di Amazon Redshift ML
- Operazione CREATE MODEL
- Funzione EXPLAIN_MODEL

Per ulteriori informazioni sul machine learning, consulta la documentazione seguente:

- Panoramica del machine learning
- Machine learning per principianti ed esperti
- Che cos'è l'equità e la spiegabilità dei modelli per le previsioni di Machine Learning?

Tutorial: Creazione di modelli XGBoost

In questo tutorial, crei un modello con i dati di Amazon S3 ed esegui query di previsione con il modello utilizzando Amazon Redshift ML. L'algoritmo XGBoost è un'implementazione ottimizzata dell'algoritmo degli alberi di gradient boosting. XGBoost gestisce più tipi di dati, relazioni e distribuzioni rispetto ad altri algoritmi basati su alberi con gradiente potenziato. È possibile utilizzare XGBoost per problemi di regressione, classificazione (binaria e multi-classe) e ordinamento. Per

ulteriori informazioni sull'algoritmo XGBoost, consulta l'algoritmo XGBoost nella Amazon Developer Guide. SageMaker

L'operazione `CREATE MODEL` di Amazon Redshift ML con l'opzione `AUTO OFF` attualmente supporta XGBoost come `MODEL_TYPE`. È possibile fornire informazioni pertinenti come l'obiettivo e gli iperparametri come parte del comando `CREATE MODEL`, in base al caso d'uso specifico.

In questo tutorial utilizzerai set di dati di autenticazione delle banconote, che è un problema di classificazione binaria, per prevedere se una determinata banconota è autentica o contraffatta.

Esempi di casi d'uso

Puoi risolvere altri problemi di classificazione binaria utilizzando Amazon Redshift ML, ad esempio prevedere se un paziente è sano o ha una malattia. Puoi anche prevedere se un'email è spam o meno.

Attività

- Prerequisiti
- Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift
- Passaggio 2: creazione del modello di machine learning
- Passaggio 3: esecuzione di previsioni con il modello

Prerequisiti

Per completare questo tutorial, è necessario completare la configurazione amministrativa di Amazon Redshift ML.

Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift

Utilizza l'editor di query v2 di Amazon Redshift per eseguire le seguenti query.

La seguente query crea due tabelle, carica i dati da Amazon S3 e suddivide i dati in un set di addestramento e un set di test. Utilizzerai il set di addestramento per addestrare il modello e creare la funzione di previsione. Testerai quindi la funzione di previsione sul set di test.

```
--create training set table
CREATE TABLE banknoteauthentication_train(
    variance FLOAT,
    skewness FLOAT,
    curtosis FLOAT,
```

```
    entropy FLOAT,
    class INT
);

--Load into training table
COPY banknoteauthentication_train
FROM
    's3://redshiftbucket-ml-sagemaker/banknote_authentication/train_data/' IAM_ROLE
 default REGION 'us-west-2' IGNOREHEADER 1 CSV;

--create testing set table
CREATE TABLE banknoteauthentication_test(
    variance FLOAT,
    skewness FLOAT,
    curtosis FLOAT,
    entropy FLOAT,
    class INT
);

--Load data into testing table
COPY banknoteauthentication_test
FROM
    's3://redshiftbucket-ml-sagemaker/banknote_authentication/test_data/'
    IAM_ROLE default
    REGION 'us-west-2'
    IGNOREHEADER 1
    CSV;
```

Passaggio 2: creazione del modello di machine learning

La seguente query crea il modello XGBoost in Amazon Redshift ML dal set di addestramento creato nel passaggio precedente. Sostituisci `DOC-EXAMPLE-BUCKET` con `S3_BUCKET` per memorizzare i set di dati di input e altri artefatti Redshift ML.

```
CREATE MODEL model_banknoteauthentication_xgboost_binary
FROM
    banknoteauthentication_train
    TARGET class
    FUNCTION func_model_banknoteauthentication_xgboost_binary
    IAM_ROLE default
    AUTO OFF
    MODEL_TYPE xgboost
    OBJECTIVE 'binary:logistic'
```

```
    PREPROCESSORS 'none'
    HYPERPARAMETERS DEFAULT
EXCEPT(NUM_ROUND '100')
SETTINGS(S3_BUCKET '<DOC-EXAMPLE-BUCKET>');
```

Visualizzazione dello stato dell'addestramento del modello (facoltativo)

È possibile utilizzare il comando SHOW MODEL per verificare quando il modello è pronto.

Utilizza la seguente query per monitorare lo stato di avanzamento dell'addestramento del modello.

```
SHOW MODEL model_banknoteauthentication_xgboost_binary;
```

Se il modello è READY, l'operazione SHOW MODEL fornisce anche il parametro `train:error`, come visualizzato nel seguente output di esempio. Il parametro `train:error` è un valore di precisione del modello contenente fino a sei cifre decimali. Un valore pari a 0 è il più preciso e un valore pari a 1 è il meno preciso.

```
+--------------------------+-------------------------------------------------+
|        Model Name        |   model_banknoteauthentication_xgboost_binary   |
+--------------------------+-------------------------------------------------+
| Schema Name              | public                                          |
| Owner                    | awsuser                                         |
| Creation Time            | Tue, 21.06.2022 19:07:35                        |
| Model State              | READY                                           |
| train:error              |                                        0.000000 |
| Estimated Cost           |                                        0.006197 |
|                          |                                                 |
| TRAINING DATA:           |                                                 |
| Query                    | SELECT *                                        |
|                          | FROM "BANKNOTEAUTHENTICATION_TRAIN"             |
| Target Column            | CLASS                                           |
|                          |                                                 |
| PARAMETERS:              |                                                 |
| Model Type               | xgboost                                         |
| Training Job Name        | redshiftml-20220621190735686935-xgboost         |
| Function Name            | func_model_banknoteauthentication_xgboost_binary |
| Function Parameters      | variance skewness curtosis entropy              |
| Function Parameter Types | float8 float8 float8 float8                     |
| IAM Role                 | default-aws-iam-role                            |
| S3 Bucket                | DOC-EXAMPLE-BUCKET    |
| Max Runtime              |                                            5400 |
```

```
|                          |                                                    |
| HYPERPARAMETERS:         |                                                    |
| num_round                |                                                100 |
| objective                | binary:logistic                                    |
+--------------------------+----------------------------------------------------+
```

Passaggio 3: esecuzione di previsioni con il modello

Verifica della precisione del modello

La seguente query di previsione utilizza la funzione di previsione creata nel passaggio precedente per verificare la precisione del modello. Esegui questa query sul set di test per assicurarti che il modello non sia eccessivamente uguale al set di addestramento. Questo tipo di corrispondenza è nota anche come overfitting; l'overfitting potrebbe far sì che il modello restituisca previsioni inaffidabili.

```
WITH predict_data AS (
    SELECT
        class AS label,
        func_model_banknoteauthentication_xgboost_binary (variance, skewness, curtosis,
 entropy) AS predicted,
        CASE
            WHEN label IS NULL THEN 0
            ELSE label
        END AS actual,
        CASE
            WHEN actual = predicted THEN 1 :: INT
            ELSE 0 :: INT
        END AS correct
    FROM
        banknoteauthentication_test
),
aggr_data AS (
    SELECT
        SUM(correct) AS num_correct,
        COUNT(*) AS total
    FROM
        predict_data
)
SELECT
    (num_correct :: FLOAT / total :: FLOAT) AS accuracy
FROM
```

```
    aggr_data;
```

Previsione della quantità di banconote originali e contraffatte

La seguente query di previsione restituisce la quantità prevista di banconote originali e contraffatte nel set di test.

```
WITH predict_data AS (
    SELECT
        func_model_banknoteauthentication_xgboost_binary(variance, skewness, curtosis,
 entropy) AS predicted
    FROM
        banknoteauthentication_test
)
SELECT
    CASE
        WHEN predicted = '0' THEN 'Original banknote'
        WHEN predicted = '1' THEN 'Counterfeit banknote'
        ELSE 'NA'
    END AS banknote_authentication,
    COUNT(1) AS count
FROM
    predict_data
GROUP BY
    1;
```

Ricerca dell'osservazione media per una banconota originale e una contraffatta

La seguente query di previsione restituisce il valore medio di ciascuna caratteristica per le banconote che si prevede siano originali e contraffatte nel set di test.

```
WITH predict_data AS (
    SELECT
        func_model_banknoteauthentication_xgboost_binary(variance, skewness, curtosis,
 entropy) AS predicted,
          variance,
          skewness,
          curtosis,
          entropy
    FROM
        banknoteauthentication_test
)
SELECT
```

```
    CASE
        WHEN predicted = '0' THEN 'Original banknote'
        WHEN predicted = '1' THEN 'Counterfeit banknote'
        ELSE 'NA'
    END AS banknote_authentication,
    TRUNC(AVG(variance), 2) AS avg_variance,
    TRUNC(AVG(skewness), 2) AS avg_skewness,
    TRUNC(AVG(curtosis), 2) AS avg_curtosis,
    TRUNC(AVG(entropy), 2) AS avg_entropy
FROM
    predict_data
GROUP BY
    1
ORDER BY
    2;
```

Argomenti correlati

Per ulteriori informazioni su Amazon Redshift ML, fare riferimento ai seguenti collegamenti:

- Costi per l'utilizzo di Amazon Redshift ML
- Operazione CREATE MODEL
- Funzione EXPLAIN_MODEL

Per ulteriori informazioni sul machine learning, consulta la documentazione seguente:

- Panoramica del machine learning
- Machine learning per principianti ed esperti
- Che cos'è l'equità e la spiegabilità dei modelli per le previsioni di Machine Learning?

Tutorial: Creazione di modelli di regressione

In questo tutorial, utilizzi Amazon Redshift ML per creare un modello di regressione basato sul machine learning ed eseguire query di previsione sul modello. I modelli di regressione consentono di prevedere risultati numerici, ad esempio il prezzo di un immobile o quante persone utilizzeranno il servizio di noleggio biciclette di una città. È possibile utilizzare il comando CREATE MODEL in Amazon Redshift con i dati di addestramento. Amazon Redshift ML compila quindi il modello, importa il modello addestrato in Redshift e prepara una funzione di previsione SQL. È possibile utilizzare la funzione di previsione nelle query SQL in Amazon Redshift.

In questo tutorial, utilizzerai Amazon Redshift ML per creare un modello di regressione che prevede il numero di persone che utilizzano il servizio di bike sharing della città di Toronto a qualsiasi ora del giorno. Gli input per il modello includono le festività e le condizioni meteorologiche. Si utilizzerà un modello di regressione, perché si desidera ottenere un risultato numerico per questo problema.

È possibile utilizzare il comando CREATE MODEL per esportare i dati di addestramento, addestrare un modello e rendere disponibile il modello in Amazon Redshift come funzione SQL. Usa l'istruzione CREATE MODEL per specificare i dati di addestramento come tabella o istruzione SELECT.

Esempi di casi d'uso

Puoi risolvere altri problemi di regressione con Amazon Redshift ML, ad esempio la previsione del valore LTV (Lifetime Value) di un cliente. Puoi anche utilizzare Redshift ML per prevedere il prezzo più remunerativo e i ricavi risultanti di un prodotto.

Attività

- Prerequisiti
- Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift
- Passaggio 2: creazione del modello di machine learning
- Passaggio 3: convalida del modello

Prerequisiti

Per completare questo tutorial, è necessario completare la configurazione amministrativa di Amazon Redshift ML.

Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift

Utilizza l'editor di query v2 di Amazon Redshift per eseguire le seguenti query.

1. È necessario creare tre tabelle per caricare i tre set di dati pubblici in Amazon Redshift. I set di dati sono Toronto Bike Ridership Data, dati cronologici meteo e dati cronologici delle festività. Esegui la seguente query nell'editor di query Amazon Redshift per creare tabelle denominate `ridership`, `weather` e `holiday`.

```
CREATE TABLE IF NOT EXISTS ridership (
    trip_id INT,
    trip_duration_seconds INT,
    trip_start_time timestamp,
```

```
    trip_stop_time timestamp,
    from_station_name VARCHAR(50),
    to_station_name VARCHAR(50),
    from_station_id SMALLINT,
    to_station_id SMALLINT,
    user_type VARCHAR(20)
);

CREATE TABLE IF NOT EXISTS weather (
    longitude_x DECIMAL(5, 2),
    latitude_y DECIMAL(5, 2),
    station_name VARCHAR(20),
    climate_id BIGINT,
    datetime_utc TIMESTAMP,
    weather_year SMALLINT,
    weather_month SMALLINT,
    weather_day SMALLINT,
    time_utc VARCHAR(5),
    temp_c DECIMAL(5, 2),
    temp_flag VARCHAR(1),
    dew_point_temp_c DECIMAL(5, 2),
    dew_point_temp_flag VARCHAR(1),
    rel_hum SMALLINT,
    rel_hum_flag VARCHAR(1),
    precip_amount_mm DECIMAL(5, 2),
    precip_amount_flag VARCHAR(1),
    wind_dir_10s_deg VARCHAR(10),
    wind_dir_flag VARCHAR(1),
    wind_spd_kmh VARCHAR(10),
    wind_spd_flag VARCHAR(1),
    visibility_km VARCHAR(10),
    visibility_flag VARCHAR(1),
    stn_press_kpa DECIMAL(5, 2),
    stn_press_flag VARCHAR(1),
    hmdx SMALLINT,
    hmdx_flag VARCHAR(1),
    wind_chill VARCHAR(10),
    wind_chill_flag VARCHAR(1),
    weather VARCHAR(10)
);

CREATE TABLE IF NOT EXISTS holiday (holiday_date DATE, description VARCHAR(100));
```

2. La seguente query consente di caricare i dati di esempio nelle tabelle create nella fase precedente.

```
COPY ridership
FROM
    's3://redshift-ml-bikesharing-data/bike-sharing-data/ridership/'
    IAM_ROLE default
    FORMAT CSV
    IGNOREHEADER 1
    DATEFORMAT 'auto'
    TIMEFORMAT 'auto'
    REGION 'us-west-2'
    gzip;

COPY weather
FROM
    's3://redshift-ml-bikesharing-data/bike-sharing-data/weather/'
    IAM_ROLE default
    FORMAT csv
    IGNOREHEADER 1
    DATEFORMAT 'auto'
    TIMEFORMAT 'auto'
    REGION 'us-west-2'
    gzip;

COPY holiday
FROM
    's3://redshift-ml-bikesharing-data/bike-sharing-data/holiday/'
    IAM_ROLE default
    FORMAT csv
    IGNOREHEADER 1
    DATEFORMAT 'auto'
    TIMEFORMAT 'auto'
    REGION 'us-west-2'
    gzip;
```

3. La seguente query esegue trasformazioni sui set di dati `ridership` e `weather` per rimuovere distorsioni o anomalie. La rimozione di distorsioni e anomalie si traduce in una migliore precisione del modello. La query semplifica le tabelle creando due nuove viste denominate `ridership_view` e `weather_view`.

```
CREATE
OR REPLACE VIEW ridership_view AS
SELECT
    trip_time,
```

```
    trip_count,
    TO_CHAR(trip_time, 'hh24') :: INT trip_hour,
    TO_CHAR(trip_time, 'dd') :: INT trip_day,
    TO_CHAR(trip_time, 'mm') :: INT trip_month,
    TO_CHAR(trip_time, 'yy') :: INT trip_year,
    TO_CHAR(trip_time, 'q') :: INT trip_quarter,
    TO_CHAR(trip_time, 'w') :: INT trip_month_week,
    TO_CHAR(trip_time, 'd') :: INT trip_week_day
FROM
    (
        SELECT
            CASE
                WHEN TRUNC(r.trip_start_time) < '2017-07-01' :: DATE THEN
 CONVERT_TIMEZONE(
                    'US/Eastern',
                    DATE_TRUNC('hour', r.trip_start_time)
                )
                ELSE DATE_TRUNC('hour', r.trip_start_time)
            END trip_time,
            COUNT(1) trip_count
        FROM
            ridership r
        WHERE
            r.trip_duration_seconds BETWEEN 60
            AND 60 * 60 * 24
        GROUP BY
            1
    );

CREATE
OR REPLACE VIEW weather_view AS
SELECT
    CONVERT_TIMEZONE(
        'US/Eastern',
        DATE_TRUNC('hour', datetime_utc)
    ) daytime,
    ROUND(AVG(temp_c)) temp_c,
    ROUND(AVG(precip_amount_mm)) precip_amount_mm
FROM
    weather
GROUP BY
    1;
```

4. La seguente query crea una tabella che combina tutti gli attributi di input rilevanti da `ridership_view` e `weather_view` nella tabella `trip_data`.

```
CREATE TABLE trip_data AS
SELECT
    r.trip_time,
    r.trip_count,
    r.trip_hour,
    r.trip_day,
    r.trip_month,
    r.trip_year,
    r.trip_quarter,
    r.trip_month_week,
    r.trip_week_day,
    w.temp_c,
    w.precip_amount_mm,CASE
        WHEN h.holiday_date IS NOT NULL THEN 1
        WHEN TO_CHAR(r.trip_time, 'D') :: INT IN (1, 7) THEN 1
        ELSE 0
    END is_holiday,
    ROW_NUMBER() OVER (
        ORDER BY
            RANDOM()
    ) serial_number
FROM
    ridership_view r
    JOIN weather_view w ON (r.trip_time = w.daytime)
    LEFT OUTER JOIN holiday h ON (TRUNC(r.trip_time) = h.holiday_date);
```

Visualizzazione dei dati di esempio (facoltativo)

La seguente query mostra le voci della tabella. È possibile eseguire questa operazione per verificare che la tabella sia stata creata correttamente.

```
SELECT *
FROM trip_data
LIMIT 5;
```

Di seguito è riportato un esempio di output dell'operazione precedente.

```
+--------------------+-----------+----------+---------+-----------+----------
+-------------+----------------+--------------+--------+-----------------
+------------+--------------+
|      trip_time      | trip_count | trip_hour | trip_day | trip_month | trip_year
 | trip_quarter | trip_month_week | trip_week_day | temp_c | precip_amount_mm |
 is_holiday | serial_number |
+--------------------+-----------+----------+---------+-----------+----------
+-------------+----------------+--------------+--------+-----------------
+------------+--------------+
| 2017-03-21 22:00:00 |         47 |        22 |       21 |          3 |        17 |
         1 |               3 |             3 |      1 |               0 |          0 |
             1 |
| 2018-05-04 01:00:00 |         19 |         1 |        4 |          5 |        18 |
         2 |               1 |             6 |     12 |               0 |          0 |
             3 |
| 2018-01-11 10:00:00 |         93 |        10 |       11 |          1 |        18 |
         1 |               2 |             5 |      9 |               0 |          0 |
             5 |
| 2017-10-28 04:00:00 |         20 |         4 |       28 |         10 |        17 |
         4 |               4 |             7 |     11 |               0 |          1 |
             7 |
| 2017-12-31 21:00:00 |         11 |        21 |       31 |         12 |        17 |
         4 |               5 |             1 |    -15 |               0 |          1 |
             9 |
+--------------------+-----------+----------+---------+-----------+----------
+-------------+----------------+--------------+--------+-----------------
+------------+--------------+
```

Visualizzazione della correlazione tra gli attributi (facoltativo)

La determinazione della correlazione aiuta a misurare la forza dell'associazione tra gli attributi. Il livello di associazione può aiutarti a determinare cosa influisce sull'output di destinazione. In questo tutorial, l'output di destinazione è `trip_count`.

La query seguente crea o sostituisce la stored procedure `sp_correlation`. Si utilizza la stored procedure denominata `sp_correlation` per mostrare la correlazione tra un attributo e altri attributi in una tabella in Amazon Redshift.

```
CREATE OR REPLACE PROCEDURE sp_correlation(source_schema_name in varchar(255),
 source_table_name in varchar(255), target_column_name in varchar(255),
 output_temp_table_name inout varchar(255)) AS $$
DECLARE
```

```
  v_sql varchar(max);
  v_generated_sql varchar(max);
  v_source_schema_name varchar(255)=lower(source_schema_name);
  v_source_table_name varchar(255)=lower(source_table_name);
  v_target_column_name varchar(255)=lower(target_column_name);
BEGIN
  EXECUTE 'DROP TABLE IF EXISTS ' || output_temp_table_name;
  v_sql = '
SELECT
  ''CREATE temp table '|| output_temp_table_name||' AS SELECT ''|| outer_calculation||
  '' FROM (SELECT COUNT(1) number_of_items, SUM('||v_target_column_name||')
 sum_target, SUM(POW('||v_target_column_name||',2)) sum_square_target, POW(SUM('||
v_target_column_name||'),2) square_sum_target,''||
  inner_calculation||
  '' FROM (SELECT ''||
  column_name||
  '' FROM '||v_source_table_name||''))''
FROM
  (
  SELECT
    DISTINCT
    LISTAGG(outer_calculation,'','') OVER () outer_calculation
    ,LISTAGG(inner_calculation,'','') OVER () inner_calculation
    ,LISTAGG(column_name,'','') OVER () column_name
  FROM
    (
    SELECT
      CASE WHEN atttypid=16 THEN ''DECODE(''||column_name||'',true,1,0)'' ELSE
 column_name END column_name
      ,atttypid
      ,''CAST(DECODE(number_of_items * sum_square_''||rn||'' - square_sum_''||
rn||'',0,null,(number_of_items*sum_target_''||rn||'' - sum_target * sum_''||rn||
        '')/SQRT((number_of_items * sum_square_target - square_sum_target) *
 (number_of_items * sum_square_''||rn||
        '' - square_sum_''||rn||''))) AS numeric(5,2)) ''||column_name
 outer_calculation
      ,''sum(''||column_name||'') sum_''||rn||'',''||
            ''SUM(trip_count*''||column_name||'') sum_target_''||rn||'',''||
            ''SUM(POW(''||column_name||'',2)) sum_square_''||rn||'',''||
            ''POW(SUM(''||column_name||''),2) square_sum_''||rn inner_calculation
    FROM
      (
      SELECT
        row_number() OVER (order by a.attnum) rn
```

```
        ,a.attname::VARCHAR column_name
        ,a.atttypid
      FROM pg_namespace AS n
        INNER JOIN pg_class AS c ON n.oid = c.relnamespace
        INNER JOIN pg_attribute AS a ON c.oid = a.attrelid
      WHERE a.attnum > 0
        AND n.nspname = '''||v_source_schema_name||'''
        AND c.relname = '''||v_source_table_name||'''
        AND a.atttypid IN (16,20,21,23,700,701,1700)
      )
    )
)';
  EXECUTE v_sql INTO v_generated_sql;
  EXECUTE  v_generated_sql;
END;
$$ LANGUAGE plpgsql;
```

La seguente query mostra la correlazione tra la colonna di destinazione `trip_count` e altri attributi numerici nel set di dati.

```
call sp_correlation(
    'public',
    'trip_data',
    'trip_count',
    'tmp_corr_table'
);

SELECT
    *
FROM
    tmp_corr_table;
```

Di seguito è illustrato un esempio di output della precedente operazione `sp_correlation`.

```
+------------+-----------+----------+------------+-----------+-------------
+-----------------+---------------+--------+------------------+------------
+---------------+
| trip_count | trip_hour | trip_day | trip_month | trip_year | trip_quarter
 | trip_month_week | trip_week_day | temp_c | precip_amount_mm | is_holiday |
 serial_number |
+------------+-----------+----------+------------+-----------+-------------
+-----------------+---------------+--------+------------------+------------
+---------------+
```

```
|          1 |      0.32 |     0.01 |       0.18 |      0.12 |         0.18 |
     0 |          0.02 |   0.53 |            -0.07 |      -0.13 |             0 |
+------------+-----------+----------+------------+-----------+--------------
+-----------------+---------------+--------+-----------------+------------
+---------------+
```

Passaggio 2: creazione del modello di machine learning

1. La seguente query suddivide i dati in un set di addestramento e in un set di convalida assegnando l'80% del set di dati per l'addestramento e il 20% per la convalida. Il set di addestramento è l'input per il modello ML per identificare il miglior algoritmo possibile per il modello. Dopo aver creato il modello, utilizza il set di convalida per convalidare l'accuratezza del modello.

```
CREATE TABLE training_data AS
SELECT
    trip_count,
    trip_hour,
    trip_day,
    trip_month,
    trip_year,
    trip_quarter,
    trip_month_week,
    trip_week_day,
    temp_c,
    precip_amount_mm,
    is_holiday
FROM
    trip_data
WHERE
    serial_number > (
        SELECT
            COUNT(1) * 0.2
        FROM
            trip_data
    );

CREATE TABLE validation_data AS
SELECT
    trip_count,
    trip_hour,
    trip_day,
    trip_month,
    trip_year,
```

```
    trip_quarter,
    trip_month_week,
    trip_week_day,
    temp_c,
    precip_amount_mm,
    is_holiday,
    trip_time
FROM
    trip_data
WHERE
    serial_number <= (
        SELECT
            COUNT(1) * 0.2
        FROM
            trip_data
    );
```

2. La seguente query crea un modello di regressione per prevedere il valore `trip_count` per qualsiasi data e ora di input. Nell'esempio seguente, sostituisci *DOC-EXAMPLE-BUCKET* con il bucket S3.

```
CREATE MODEL predict_rental_count
FROM
    training_data TARGET trip_count FUNCTION predict_rental_count
    IAM_ROLE default
    PROBLEM_TYPE regression
    OBJECTIVE 'mse'
    SETTINGS (
        s3_bucket '<DOC-EXAMPLE-BUCKET>',
        s3_garbage_collect off,
        max_runtime 5000
    );
```

Passaggio 3: convalida del modello

1. Utilizzare la seguente query per generare l'output relativo agli aspetti del modello e per trovare il parametro relativo all'errore quadratico medio nell'output. L'errore quadratico medio è un parametro di precisione standard per i problemi di regressione.

```
show model predict_rental_count;
```

2. Eseguire le seguenti query di previsione sui dati di convalida per confrontare il conteggio previsto con il conteggio effettivo dei viaggi.

```
SELECT
    trip_time,
    actual_count,
    predicted_count,
    (actual_count - predicted_count) difference
FROM
    (
        SELECT
            trip_time,
            trip_count AS actual_count,
            PREDICT_RENTAL_COUNT (
                trip_hour,
                trip_day,
                trip_month,
                trip_year,
                trip_quarter,
                trip_month_week,
                trip_week_day,
                temp_c,
                precip_amount_mm,
                is_holiday
            ) predicted_count
        FROM
            validation_data
    )
LIMIT
    5;
```

3. La seguente query calcola l'errore quadratico medio e la relativa radice in base ai dati di convalida. Utilizzare l'errore quadratico medio e la relativa radice per misurare la distanza tra il target numerico previsto e la risposta numerica effettiva. Un modello ottimale è caratterizzato da un punteggio basso per entrambi i parametri. La seguente query restituisce il valore di entrambi i parametri.

```
SELECT
    ROUND(
        AVG(POWER((actual_count - predicted_count), 2)),
        2
    ) mse,
```

```
    ROUND(
        SQRT(AVG(POWER((actual_count - predicted_count), 2))),
        2
    ) rmse
FROM
    (
        SELECT
            trip_time,
            trip_count AS actual_count,
            PREDICT_RENTAL_COUNT (
                trip_hour,
                trip_day,
                trip_month,
                trip_year,
                trip_quarter,
                trip_month_week,
                trip_week_day,
                temp_c,
                precip_amount_mm,
                is_holiday
            ) predicted_count
        FROM
            validation_data
    );
```

4. La seguente query calcola l'errore percentuale nel conteggio dei viaggi per ogni durata del viaggio per la data 01/01/2017. La query ordina i tempi di viaggio dal momento con l'errore percentuale più basso a quello con la percentuale di errore più alta.

```
SELECT
    trip_time,
    CAST(ABS(((actual_count - predicted_count) / actual_count)) * 100 AS DECIMAL
 (7,2)) AS pct_error
FROM
    (
        SELECT
            trip_time,
            trip_count AS actual_count,
            PREDICT_RENTAL_COUNT (
                trip_hour,
                trip_day,
                trip_month,
                trip_year,
```

```
                trip_quarter,
                trip_month_week,
                trip_week_day,
                temp_c,
                precip_amount_mm,
                is_holiday
            ) predicted_count
        FROM
            validation_data
    )
WHERE
   trip_time LIKE '2017-01-01 %%:%%:%%'
ORDER BY
   2 ASC;
```

Argomenti correlati

Per ulteriori informazioni su Amazon Redshift ML, fare riferimento ai seguenti collegamenti:

- Costi per l'utilizzo di Amazon Redshift ML
- Operazione CREATE MODEL
- Funzione EXPLAIN_MODEL

Per ulteriori informazioni sul machine learning, consulta la documentazione seguente:

- Panoramica del machine learning
- Machine learning per principianti ed esperti
- Che cos'è l'equità e la spiegabilità dei modelli per le previsioni di Machine Learning?

Tutorial: Creazione di modelli di regressione con Linear Learner

In questo tutorial, crei un modello basato sull'algoritmo Linear Learner con i dati di Amazon S3 ed esegui query di previsione con il modello utilizzando Amazon Redshift ML. L'algoritmo di apprendimento SageMaker lineare risolve problemi di regressione o di classificazione multiclasse. Per ulteriori informazioni sulla regressione e sui problemi di classificazione multiclasse, consulta Tipi di problemi per i paradigmi di apprendimento automatico nella Amazon Developer Guide. SageMaker In questo tutorial, risolverai un problema di regressione. L'algoritmo Linear Learner addestra molti modelli in parallelo e determina automaticamente il modello più ottimizzato. Utilizzi l'operazione

CREATE MODEL in Amazon Redshift, che crea il tuo modello lineare di apprendimento utilizzando SageMaker e invia una funzione di previsione ad Amazon Redshift. Per ulteriori informazioni sull'algoritmo Linear Learner, consulta Linear Learner Algorithm nella Amazon SageMaker Developer Guide.

È possibile usare un comando CREATE MODEL per esportare i dati di addestramento, addestrare un modello, importare il modello e preparare una funzione di previsione Amazon Redshift. Usa l'istruzione CREATE MODEL per specificare i dati di addestramento come tabella o istruzione SELECT.

I modelli Linear Learner ottimizzano obiettivi continui o obiettivi discreti. Gli obiettivi continui vengono utilizzati per la regressione, mentre le variabili discrete vengono utilizzate per la classificazione. Alcuni metodi forniscono una soluzione solo per obiettivi continui, come il metodo di regressione. L'algoritmo Linear Learner garantisce un incremento della velocità con le tecniche di ottimizzazione di tipo Naive degli iperparametri, ad esempio la tecnica Naive Bayes. Una tecnica di ottimizzazione di tipo Naive presuppone che ogni variabile di input sia indipendente. Per utilizzare l'algoritmo Linear Learner, è necessario specificare colonne che rappresentano le dimensioni degli input e righe che rappresentano le osservazioni. Per ulteriori informazioni sull'algoritmo Linear Learner, consulta il Linear Learner Algorithm nella Amazon SageMaker Developer Guide.

In questo tutorial, costruisci un modello Linear Learner che prevede l'età dell'abalone. Utilizzerai il comando CREATE MODEL con il set di dati relativo agli abaloni per determinare la relazione tra le misure fisiche dell'abalone. Userai quindi il modello per determinare l'età dell'abalone.

Esempi di casi d'uso

Puoi risolvere altri problemi di regressione con la funzionalità Linear Learner e Amazon Redshift ML, come la previsione del prezzo di un immobile. Puoi anche utilizzare Redshift ML per prevedere il numero di persone che utilizzeranno il servizio di noleggio di biciclette di una città.

Attività

- Prerequisiti
- Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift
- Passaggio 2: creazione del modello di machine learning
- Passaggio 3: convalida del modello

Prerequisiti

Per completare questo tutorial, è necessario completare la configurazione amministrativa di Amazon Redshift ML.

Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift

Utilizza l'editor di query v2 di Amazon Redshift per eseguire le seguenti query. Queste query caricano i dati di esempio in Redshift e li suddividono in un set di addestramento e un set di convalida.

1. La query seguente crea una tabella `abalone_dataset`.

```
CREATE TABLE abalone_dataset (
    id INT IDENTITY(1, 1),
    Sex CHAR(1),
    Length float,
    Diameter float,
    Height float,
    Whole float,
    Shucked float,
    Viscera float,
    Shell float,
    Rings integer
);
```

2. La seguente query copia i dati di esempio dal set di dati relativo agli abaloni in Amazon S3 nella tabella `abalone_dataset` creata in precedenza in Amazon Redshift.

```
COPY abalone_dataset
FROM
    's3://redshift-ml-multiclass/abalone.csv' REGION 'us-east-1' IAM_ROLE default CSV
 IGNOREHEADER 1 NULL AS 'NULL';
```

3. Suddividendo manualmente i dati, sarai in grado di verificare l'accuratezza del modello assegnando un set di previsione aggiuntivo. La seguente query suddivide i dati in due set. La tabella `abalone_training` è destinata all'allenamento, mentre la tabella `abalone_validation` è destinata alla convalida.

```
CREATE TABLE abalone_training as
SELECT
    *
FROM
```

```
    abalone_dataset
WHERE
    mod(id, 10) < 8;

CREATE TABLE abalone_validation as
SELECT
    *
FROM
    abalone_dataset
WHERE
    mod(id, 10) >= 8;
```

Passaggio 2: creazione del modello di machine learning

In questo passaggio, si utilizza l'istruzione CREATE MODEL per creare il modello di machine learning con l'algoritmo Linear Learner.

La seguente query crea il modello Linear Learner con l'operazione CREATE MODEL utilizzando il bucket S3. Sostituisci *DOC-EXAMPLE-BUCKET* con il bucket S3.

```
CREATE MODEL model_abalone_ring_prediction
FROM
    (
        SELECT
            Sex,
            Length,
            Diameter,
            Height,
            Whole,
            Shucked,
            Viscera,
            Shell,
            Rings AS target_label
        FROM
            abalone_training
    ) TARGET target_label FUNCTION f_abalone_ring_prediction IAM_ROLE default
 MODEL_TYPE LINEAR_LEARNER PROBLEM_TYPE REGRESSION OBJECTIVE 'MSE' SETTINGS (
        S3_BUCKET 'DOC-EXAMPLE-BUCKET',
        MAX_RUNTIME 15000
    );
```

Visualizzazione dello stato dell'addestramento del modello (facoltativo)

È possibile utilizzare il comando SHOW MODEL per verificare quando il modello è pronto.

Utilizza la seguente query per monitorare lo stato di avanzamento dell'addestramento del modello.

```
SHOW MODEL model_abalone_ring_prediction;
```

Quando il modello è pronto, l'output dell'operazione precedente dovrebbe essere simile all'esempio seguente. Nota: l'output riporta il parametro `validation:mse`, che è l'errore quadratico medio. Userai un errore quadratico medio per convalidare la precisione del modello nella fase successiva.

```
+--------------------------
+-----------------------------------------------------------------------------------------------
+
|        Model Name        |
 model_abalone_ring_prediction                                    |
+--------------------------
+-----------------------------------------------------------------------------------------------
+
| Schema Name              | public
                                        |
| Owner                    | awsuser
                                        |
| Creation Time            | Thu, 30.06.2022 18:00:10
                                        |
| Model State              | READY
                                        |
| validation:mse           |
                                 4.168633 |
| Estimated Cost           |
                                 4.291608 |
|                          |
                                        |
| TRAINING DATA:           |
                                        |
| Query                    | SELECT SEX , LENGTH , DIAMETER , HEIGHT , WHOLE ,
 SHUCKED , VISCERA , SHELL, RINGS AS TARGET_LABEL |
|                          | FROM ABALONE_TRAINING
                                        |
| Target Column            | TARGET_LABEL
                                        |
```

```
|                           |
                                        |
| PARAMETERS:               |
                                        |
| Model Type                | linear_learner
                                        |
| Problem Type              | Regression
                                        |
| Objective                 | MSE
                                        |
| AutoML Job Name           | redshiftml-20220630180010947843
                                        |
| Function Name             | f_abalone_ring_prediction
                                        |
| Function Parameters       | sex length diameter height whole shucked viscera shell
                                        |
| Function Parameter Types | bpchar float8 float8 float8 float8 float8 float8 float8
                                        |
| IAM Role                  | default-aws-iam-role
                                        |
| S3 Bucket                 | DOC-EXAMPLE-BUCKET
                                        |
| Max Runtime               |
                                15000 |
+--------------------------
+-------------------------------------------------------------------------------------------------
+
```

Passaggio 3: convalida del modello

1. La seguente query di previsione convalida l'accuratezza del modello nel set di dati `abalone_validation` calcolando l'errore quadratico medio e la relativa radice.

```
SELECT
    ROUND(AVG(POWER((tgt_label - predicted), 2)), 2) mse,
    ROUND(SQRT(AVG(POWER((tgt_label - predicted), 2))), 2) rmse
FROM
    (
        SELECT
            Sex,
            Length,
            Diameter,
            Height,
```

```
            Whole,
            Shucked,
            Viscera,
            Shell,
            Rings AS tgt_label,
            f_abalone_ring_prediction(
                Sex,
                Length,
                Diameter,
                Height,
                Whole,
                Shucked,
                Viscera,
                Shell
            ) AS predicted,
            CASE
                WHEN tgt_label = predicted then 1
                ELSE 0
            END AS match,
            CASE
                WHEN tgt_label <> predicted then 1
                ELSE 0
            END AS nonmatch
        FROM
            abalone_validation
    ) t1;
```

L'output della precedente query dovrebbe essere simile all'output di esempio seguente. Il valore del parametro relativo all'errore quadratico medio deve essere simile al parametro `validation:mse` visualizzato nell'output dell'operazione SHOW MODEL.

```
+-----+--------------------+
| mse |        rmse        |
+-----+--------------------+
| 5.1 | 2.2600000000000002 |
+-----+--------------------+
```

2. Utilizzare la seguente query per eseguire l'operazione EXPLAIN_MODEL sulla funzione di previsione. L'operazione restituirà un report sulla spiegabilità del modello. Per ulteriori informazioni sull'operazione EXPLAIN_MODEL, consultare Funzione EXPLAIN_MODEL nella Guida per sviluppatori di database di Amazon Redshift.

```
SELECT
    EXPLAIN_MODEL ('model_abalone_ring_prediction');
```

Le informazioni seguenti sono un esempio del report sulla spiegabilità del modello generato dalla precedente operazione EXPLAIN_MODEL. I valori per ciascuno degli input sono valori di Shapley. I valori di Shapley rappresentano l'effetto che ogni input ha sulla previsione del modello, con input di valore più elevato che hanno un impatto maggiore sulla previsione. In questo esempio, gli input di valore più elevato hanno un impatto maggiore sulla previsione dell'età dell'abalone.

```
{
    "explanations": {
        "kernel_shap": {
            "label0": {
                "expected_value" :10.290688514709473,
                "global_shap_values": {
                    "diameter" :0.6856910187882492,
                    "height" :0.4415323937124035,
                    "length" :0.21507476107609084,
                    "sex" :0.448611774505744,
                    "shell" :1.70426496893776,
                    "shucked" :2.1181392924386994,
                    "viscera" :0.342220754059912,
                    "whole" :0.6711906974084011
                }
            }
        }
    },
    "version" :"1.0"
};
```

3. Utilizzare la seguente query per calcolare la percentuale di previsioni corrette che il modello genera sugli abaloni non ancora maturi. Gli abaloni immaturi hanno 10 anelli o meno; una previsione corretta è accurata con un anello di scarto rispetto al numero effettivo di anelli.

```
SELECT
    TRUNC(
        SUM(
            CASE
                WHEN ROUND(
                    f_abalone_ring_prediction(
```

```
                        Sex,
                        Length,
                        Diameter,
                        Height,
                        Whole,
                        Shucked,
                        Viscera,
                        Shell
                    ),
                    0
                ) BETWEEN Rings - 1
                AND Rings + 1 THEN 1
                ELSE 0
            END
        ) / CAST(COUNT(SHELL) AS FLOAT),
        4
    ) AS prediction_pct
FROM
    abalone_validation
WHERE
    Rings <= 10;
```

Argomenti correlati

Per ulteriori informazioni su Amazon Redshift ML, fare riferimento ai seguenti collegamenti:

- Costi per l'utilizzo di Amazon Redshift ML
- Operazione CREATE MODEL
- Funzione EXPLAIN_MODEL

Per ulteriori informazioni sul machine learning, consulta la documentazione seguente:

- Panoramica del machine learning
- Machine learning per principianti ed esperti
- Che cos'è l'equità e la spiegabilità dei modelli per le previsioni di Machine Learning?

Tutorial: Creazione di modelli di classificazione multi-classe con Linear Learner

In questo tutorial, crei un modello Linear Learner con i dati di Amazon S3, quindi esegui query di previsione con il modello utilizzando Amazon Redshift ML. L'algoritmo SageMaker lineare di apprendimento risolve problemi di regressione o classificazione. Per ulteriori informazioni sulla regressione e sui problemi di classificazione multiclasse, consulta Tipi di problemi per i paradigmi di apprendimento automatico nella Amazon Developer Guide. SageMaker In questo tutorial, risolverai un problema di classificazione multi-classe. L'algoritmo Linear Learner addestra molti modelli in parallelo e determina automaticamente il modello più ottimizzato. Utilizzi l'operazione CREATE MODEL in Amazon Redshift, che crea il tuo modello lineare di apprendimento utilizzando SageMaker e invia la funzione di previsione ad Amazon Redshift. Per ulteriori informazioni sull'algoritmo Linear Learner, consulta il Linear Learner Algorithm nella Amazon SageMaker Developer Guide.

È possibile usare un comando CREATE MODEL per esportare i dati di addestramento, addestrare un modello, importare il modello e preparare una funzione di previsione Amazon Redshift. Usa l'istruzione CREATE MODEL per specificare i dati di addestramento come tabella o istruzione SELECT.

I modelli Linear Learner ottimizzano obiettivi continui o obiettivi discreti. Gli obiettivi continui vengono utilizzati per la regressione, mentre le variabili discrete vengono utilizzate per la classificazione. Alcuni metodi forniscono una soluzione solo per obiettivi continui, ad esempio un metodo di regressione. L'algoritmo Linear Learner garantisce un incremento della velocità con le tecniche di ottimizzazione di tipo Naive degli iperparametri, ad esempio la tecnica Naive Bayes. Una tecnica di ottimizzazione di tipo Naive presuppone che ogni variabile di input sia indipendente. L'algoritmo Linear Learner addestra molti modelli in parallelo e seleziona il modello più ottimizzato. Un algoritmo simile è XGBoost, che combina le stime di una serie di modelli più semplici e più deboli per la generazione di previsioni. Per ulteriori informazioni su XGBoost, consulta l'algoritmo XGBoost nella Amazon Developer Guide. SageMaker

Per utilizzare l'algoritmo Linear Learner, è necessario specificare colonne che rappresentano le dimensioni degli input e righe che rappresentano le osservazioni. Per ulteriori informazioni sull'algoritmo Linear Learner, consulta il Linear Learner Algorithm nella Amazon SageMaker Developer Guide.

In questo tutorial, costruisci un modello Linear Learner che prevede i tipi di copertura per una determinata area. Userai il comando CREATE MODEL con il set di dati relativo al tipo di copertura disponibile nella pagina UCI Machine Learning Repository. Utilizzerai quindi la funzione di previsione creata dal comando per determinare i tipi di copertura in un'area naturale. Un tipo di copertura forestale è generalmente un tipo di albero. Gli input che Redshift ML utilizzerà per creare il modello

includono il tipo di suolo, la distanza dalle strade e la designazione delle aree naturali. Per ulteriori informazioni sul set di dati, consulta il set di dati relativo al tipo di copertura disponibile nella pagina UCI Machine Learning Repository.

Esempi di casi d'uso

Puoi risolvere altri problemi di classificazione multi-classe con il modello Linear Learner di Amazon Redshift ML, ad esempio la previsione della specie di una pianta in base a un'immagine. Puoi anche prevedere la quantità di un prodotto che un cliente acquisterà.

Attività

- Prerequisiti
- Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift
- Passaggio 2: creazione del modello di machine learning
- Passaggio 3: convalida del modello

Prerequisiti

Per completare questo tutorial, è necessario completare la configurazione amministrativa di Amazon Redshift ML.

Passaggio 1: caricamento dei dati da Amazon S3 ad Amazon Redshift

Utilizza l'editor di query v2 di Amazon Redshift per eseguire le seguenti query. Queste query caricano i dati di esempio in Redshift e li suddividono in un set di addestramento e un set di convalida.

1. La query seguente crea una tabella `covertype_data`.

```
CREATE TABLE public.covertype_data (
    elevation bigint ENCODE az64,
    aspect bigint ENCODE az64,
    slope bigint ENCODE az64,
    horizontal_distance_to_hydrology bigint ENCODE az64,
    vertical_distance_to_hydrology bigint ENCODE az64,
    horizontal_distance_to_roadways bigint ENCODE az64,
    hillshade_9am bigint ENCODE az64,
    hillshade_noon bigint ENCODE az64,
    hillshade_3pm bigint ENCODE az64,
    horizontal_distance_to_fire_points bigint ENCODE az64,
    wilderness_area1 bigint ENCODE az64,
```

```
    wilderness_area2 bigint ENCODE az64,
    wilderness_area3 bigint ENCODE az64,
    wilderness_area4 bigint ENCODE az64,
    soil_type1 bigint ENCODE az64,
    soil_type2 bigint ENCODE az64,
    soil_type3 bigint ENCODE az64,
    soil_type4 bigint ENCODE az64,
    soil_type5 bigint ENCODE az64,
    soil_type6 bigint ENCODE az64,
    soil_type7 bigint ENCODE az64,
    soil_type8 bigint ENCODE az64,
    soil_type9 bigint ENCODE az64,
    soil_type10 bigint ENCODE az64,
    soil_type11 bigint ENCODE az64,
    soil_type12 bigint ENCODE az64,
    soil_type13 bigint ENCODE az64,
    soil_type14 bigint ENCODE az64,
    soil_type15 bigint ENCODE az64,
    soil_type16 bigint ENCODE az64,
    soil_type17 bigint ENCODE az64,
    soil_type18 bigint ENCODE az64,
    soil_type19 bigint ENCODE az64,
    soil_type20 bigint ENCODE az64,
    soil_type21 bigint ENCODE az64,
    soil_type22 bigint ENCODE az64,
    soil_type23 bigint ENCODE az64,
    soil_type24 bigint ENCODE az64,
    soil_type25 bigint ENCODE az64,
    soil_type26 bigint ENCODE az64,
    soil_type27 bigint ENCODE az64,
    soil_type28 bigint ENCODE az64,
    soil_type29 bigint ENCODE az64,
    soil_type30 bigint ENCODE az64,
    soil_type31 bigint ENCODE az64,
    soil_type32 bigint ENCODE az64,
    soil_type33 bigint ENCODE az64,
    soil_type34 bigint ENCODE az64,
    soil_type35 bigint ENCODE az64,
    soil_type36 bigint ENCODE az64,
    soil_type37 bigint ENCODE az64,
    soil_type38 bigint ENCODE az64,
    soil_type39 bigint ENCODE az64,
    soil_type40 bigint ENCODE az64,
    cover_type bigint ENCODE az64
```

```
) DISTSTYLE AUTO;
```

2. La seguente query copia i dati di esempio dal set di dati relativo al tipo di copertura in Amazon S3 nella tabella `covertype_data` creata in precedenza in Amazon Redshift.

```
COPY public.covertype_data
FROM
    's3://redshift-ml-multiclass/covtype.data.gz' IAM_ROLE DEFAULT gzip DELIMITER ','
 REGION 'us-east-1';
```

3. Dividendo manualmente i dati, sarai in grado di verificare l'accuratezza del modello assegnando un set di test aggiuntivo. La seguente query suddivide i dati in tre set. La tabella `covertype_training` è destinata all'addestramento, la tabella `covertype_validation` alla convalida e la tabella `covertype_test` al test del modello. Userai il set di addestramento per addestrare il modello e il set di convalida per convalidare lo sviluppo del modello. Userai quindi il set di test per testare le prestazioni del modello e verificare se si è verificato l'overfitting o l'underfitting del modello rispetto al set di dati.

```
CREATE TABLE public.covertype_data_prep AS
SELECT
    a.*,
    CAST (random() * 100 AS int) AS data_group_id
FROM
    public.covertype_data a;

--training dataset
 CREATE TABLE public.covertype_training as
SELECT
    *
FROM
    public.covertype_data_prep
WHERE
    data_group_id < 80;

--validation dataset
 CREATE TABLE public.covertype_validation AS
SELECT
    *
FROM
    public.covertype_data_prep
WHERE
    data_group_id BETWEEN 80
```

```
    AND 89;

--test dataset
 CREATE TABLE public.covertype_test AS
SELECT
    *
FROM
    public.covertype_data_prep
WHERE
    data_group_id > 89;
```

Passaggio 2: creazione del modello di machine learning

In questo passaggio, si utilizza l'istruzione CREATE MODEL per creare il modello di machine learning con l'algoritmo Linear Learner.

La seguente query crea il modello Linear Learner con l'operazione CREATE MODEL utilizzando il bucket S3. Sostituisci *DOC-EXAMPLE-BUCKET* con il bucket S3.

```
CREATE MODEL forest_cover_type_model
FROM
    (
        SELECT
            Elevation,
            Aspect,
            Slope,
            Horizontal_distance_to_hydrology,
            Vertical_distance_to_hydrology,
            Horizontal_distance_to_roadways,
            HIllshade_9am,
            Hillshade_noon,
            Hillshade_3pm,
            Horizontal_Distance_To_Fire_Points,
            Wilderness_Area1,
            Wilderness_Area2,
            Wilderness_Area3,
            Wilderness_Area4,
            soil_type1,
            Soil_Type2,
            Soil_Type3,
            Soil_Type4,
            Soil_Type5,
```

```
            Soil_Type6,
            Soil_Type7,
            Soil_Type8,
            Soil_Type9,
            Soil_Type10,
            Soil_Type11,
            Soil_Type12,
            Soil_Type13,
            Soil_Type14,
            Soil_Type15,
            Soil_Type16,
            Soil_Type17,
            Soil_Type18,
            Soil_Type19,
            Soil_Type20,
            Soil_Type21,
            Soil_Type22,
            Soil_Type23,
            Soil_Type24,
            Soil_Type25,
            Soil_Type26,
            Soil_Type27,
            Soil_Type28,
            Soil_Type29,
            Soil_Type30,
            Soil_Type31,
            Soil_Type32,
            Soil_Type33,
            Soil_Type34,
            Soil_Type36,
            Soil_Type37,
            Soil_Type38,
            Soil_Type39,
            Soil_Type40,
            Cover_type
        from
            public.covertype_training
    ) TARGET cover_type FUNCTION predict_cover_type IAM_ROLE default MODEL_TYPE
LINEAR_LEARNER PROBLEM_TYPE MULTICLASS_CLASSIFICATION OBJECTIVE 'Accuracy' SETTINGS (
        S3_BUCKET '<DOC-EXAMPLE-BUCKET>',
        S3_GARBAGE_COLLECT OFF,
        MAX_RUNTIME 15000
    );
```

Visualizzazione dello stato dell'addestramento del modello (facoltativo)

È possibile utilizzare il comando SHOW MODEL per verificare quando il modello è pronto.

Utilizza la seguente query per monitorare lo stato di avanzamento dell'addestramento del modello.

```
SHOW MODEL forest_cover_type_model;
```

Quando il modello è pronto, l'output dell'operazione precedente dovrebbe essere simile all'esempio seguente. Nota: l'output riporta il parametro `validation:multiclass_accuracy`, visibile sul lato destro del seguente esempio. La precisione multi-classe misura la percentuale di punti dati classificati correttamente dal modello. Utilizzerai la precisione multi-classe per convalidare l'accuratezza del modello nel passaggio successivo.

```
+---------------------------------
+-----------------------------------------------------------------------------------------------
+
|              Key               |




 Value




                                                        |
+---------------------------------
+-----------------------------------------------------------------------------------------------
+
| Model Name                     | forest_cover_type_model









                                                      |
| Schema Name                    | public
```

```
                                                    |
| Owner                          | awsuser
                                                    |
| Creation Time                  | Tue, 12.07.2022 20:24:32
                                                    |
| Model State                    | READY
                                                    |
| validation:multiclass_accuracy |
```

```
                                        0.724952 |
| Estimated Cost                 |

                                        5.341750 |
|                                |

                                                 |
| TRAINING DATA:                 |

                                                 |
| Query                          | SELECT ELEVATION, ASPECT, SLOPE,
 HORIZONTAL_DISTANCE_TO_HYDROLOGY, VERTICAL_DISTANCE_TO_HYDROLOGY,
 HORIZONTAL_DISTANCE_TO_ROADWAYS, HILLSHADE_9AM, HILLSHADE_NOON, HILLSHADE_3PM ,
 HORIZONTAL_DISTANCE_TO_FIRE_POINTS, WILDERNESS_AREA1, WILDERNESS_AREA2,
 WILDERNESS_AREA3, WILDERNESS_AREA4, SOIL_TYPE1, SOIL_TYPE2, SOIL_TYPE3, SOIL_TYPE4,
 SOIL_TYPE5, SOIL_TYPE6, SOIL_TYPE7, SOIL_TYPE8, SOIL_TYPE9, SOIL_TYPE10 , SOIL_TYPE11,
 SOIL_TYPE12 , SOIL_TYPE13 , SOIL_TYPE14, SOIL_TYPE15, SOIL_TYPE16, SOIL_TYPE17,
 SOIL_TYPE18, SOIL_TYPE19, SOIL_TYPE20, SOIL_TYPE21, SOIL_TYPE22, SOIL_TYPE23,
 SOIL_TYPE24, SOIL_TYPE25, SOIL_TYPE26, SOIL_TYPE27, SOIL_TYPE28, SOIL_TYPE29,
 SOIL_TYPE30, SOIL_TYPE31, SOIL_TYPE32, SOIL_TYPE33, SOIL_TYPE34, SOIL_TYPE36,
 SOIL_TYPE37, SOIL_TYPE38, SOIL_TYPE39, SOIL_TYPE40, COVER_TYPE |
```

```
|                                | FROM PUBLIC.COVERTYPE_TRAINING

                                                                 |
| Target Column                  | COVER_TYPE

                                                                 |
|                                |

                                                                 |
| PARAMETERS:                    |

                                                                 |
| Model Type                     | linear_learner
```

```
                                                    |
| Problem Type                    | MulticlassClassification

                                                    |
| Objective                       | Accuracy

                                                    |
| AutoML Job Name                 | redshiftml-20220712202432187659

                                                    |
| Function Name                   | predict_cover_type
```

```
                                                    |
| Function Parameters            | elevation aspect slope
 horizontal_distance_to_hydrology vertical_distance_to_hydrology
 horizontal_distance_to_roadways hillshade_9am hillshade_noon hillshade_3pm
 horizontal_distance_to_fire_points wilderness_area1 wilderness_area2 wilderness_area3
 wilderness_area4 soil_type1 soil_type2 soil_type3 soil_type4 soil_type5 soil_type6
 soil_type7 soil_type8 soil_type9 soil_type10 soil_type11 soil_type12 soil_type13
 soil_type14 soil_type15 soil_type16 soil_type17 soil_type18 soil_type19 soil_type20
 soil_type21 soil_type22 soil_type23 soil_type24 soil_type25 soil_type26 soil_type27
 soil_type28 soil_type29 soil_type30 soil_type31 soil_type32 soil_type33 soil_type34
 soil_type36 soil_type37 soil_type38 soil_type39 soil_type40
                                                   |
| Function Parameter Types       | int8 int8 int8 int8 int8 int8 int8 int8 int8 int8
 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8
 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8
 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8 int8
 int8 int8 int8 int8 int8 int8 int8 int8 int8
                                                            |
| IAM Role                       | default-aws-iam-role
                                                    |
| S3 Bucket                      | DOC-EXAMPLE-BUCKET
                                                    |
```

```
| Max Runtime                    |

                                                15000 |
+--------------------------------
+-----------------------------------------------------------------------------------------------
+
```

Passaggio 3: convalida del modello

1. La seguente query di previsione convalida l'accuratezza del modello sul set di dati `covertype_validation` mediante il calcolo della precisione multi-classe. La precisione multi-classe è la percentuale delle previsioni del modello corrette.

```
SELECT
    CAST(sum(t1.match) AS decimal(7, 2)) AS predicted_matches,
    CAST(sum(t1.nonmatch) AS decimal(7, 2)) AS predicted_non_matches,
    CAST(sum(t1.match + t1.nonmatch) AS decimal(7, 2)) AS total_predictions,
    predicted_matches / total_predictions AS pct_accuracy
FROM
    (
        SELECT
            Elevation,
            Aspect,
            Slope,
            Horizontal_distance_to_hydrology,
            Vertical_distance_to_hydrology,
            Horizontal_distance_to_roadways,
            HIllshade_9am,
            Hillshade_noon,
            Hillshade_3pm,
            Horizontal_Distance_To_Fire_Points,
            Wilderness_Area1,
            Wilderness_Area2,
            Wilderness_Area3,
            Wilderness_Area4,
            soil_type1,
```

```
            Soil_Type2,
            Soil_Type3,
            Soil_Type4,
            Soil_Type5,
            Soil_Type6,
            Soil_Type7,
            Soil_Type8,
            Soil_Type9,
            Soil_Type10,
            Soil_Type11,
            Soil_Type12,
            Soil_Type13,
            Soil_Type14,
            Soil_Type15,
            Soil_Type16,
            Soil_Type17,
            Soil_Type18,
            Soil_Type19,
            Soil_Type20,
            Soil_Type21,
            Soil_Type22,
            Soil_Type23,
            Soil_Type24,
            Soil_Type25,
            Soil_Type26,
            Soil_Type27,
            Soil_Type28,
            Soil_Type29,
            Soil_Type30,
            Soil_Type31,
            Soil_Type32,
            Soil_Type33,
            Soil_Type34,
            Soil_Type36,
            Soil_Type37,
            Soil_Type38,
            Soil_Type39,
            Soil_Type40,
            Cover_type AS actual_cover_type,
            predict_cover_type(
                Elevation,
                Aspect,
                Slope,
                Horizontal_distance_to_hydrology,
```

```
                Vertical_distance_to_hydrology,
                Horizontal_distance_to_roadways,
                HIllshade_9am,
                Hillshade_noon,
                Hillshade_3pm,
                Horizontal_Distance_To_Fire_Points,
                Wilderness_Area1,
                Wilderness_Area2,
                Wilderness_Area3,
                Wilderness_Area4,
                soil_type1,
                Soil_Type2,
                Soil_Type3,
                Soil_Type4,
                Soil_Type5,
                Soil_Type6,
                Soil_Type7,
                Soil_Type8,
                Soil_Type9,
                Soil_Type10,
                Soil_Type11,
                Soil_Type12,
                Soil_Type13,
                Soil_Type14,
                Soil_Type15,
                Soil_Type16,
                Soil_Type17,
                Soil_Type18,
                Soil_Type19,
                Soil_Type20,
                Soil_Type21,
                Soil_Type22,
                Soil_Type23,
                Soil_Type24,
                Soil_Type25,
                Soil_Type26,
                Soil_Type27,
                Soil_Type28,
                Soil_Type29,
                Soil_Type30,
                Soil_Type31,
                Soil_Type32,
                Soil_Type33,
                Soil_Type34,
```

```
                Soil_Type36,
                Soil_Type37,
                Soil_Type38,
                Soil_Type39,
                Soil_Type40
            ) AS predicted_cover_type,
            CASE
                WHEN actual_cover_type = predicted_cover_type THEN 1
                ELSE 0
            END AS match,
            CASE
                WHEN actual_cover_type <> predicted_cover_type THEN 1
                ELSE 0
            END AS nonmatch
        FROM
            public.covertype_validation
    ) t1;
```

L'output della precedente query dovrebbe essere simile all'output di esempio seguente.
Il valore del parametro di precisione multi-classe deve essere simile al parametro `validation:multiclass_accuracy` visualizzato nell'output dell'operazione SHOW MODEL.

```
+------------------+----------------------+------------------+-------------+
| predicted_matches | predicted_non_matches | total_predictions | pct_accuracy |
+------------------+----------------------+------------------+-------------+
|             41211 |                16324 |             57535 |   0.71627704 |
+------------------+----------------------+------------------+-------------+
```

2. La seguente query prevede il tipo di copertina più comune per `wilderness_area2`. Questo set di dati include quattro aree naturali e sette tipi di copertura. Un'area naturale può avere più tipi di copertura.

```
SELECT t1. predicted_cover_type, COUNT(*)
FROM
(
SELECT
   Elevation,
   Aspect,
   Slope,
   Horizontal_distance_to_hydrology,
   Vertical_distance_to_hydrology,
   Horizontal_distance_to_roadways,
```

```
    HIllshade_9am,
    Hillshade_noon,
    Hillshade_3pm ,
    Horizontal_Distance_To_Fire_Points,
    Wilderness_Area1,
    Wilderness_Area2,
    Wilderness_Area3,
    Wilderness_Area4,
    soil_type1,
    Soil_Type2,
    Soil_Type3,
    Soil_Type4,
    Soil_Type5,
    Soil_Type6,
    Soil_Type7,
    Soil_Type8,
    Soil_Type9,
    Soil_Type10 ,
    Soil_Type11,
    Soil_Type12 ,
    Soil_Type13 ,
    Soil_Type14,
    Soil_Type15,
    Soil_Type16,
    Soil_Type17,
    Soil_Type18,
    Soil_Type19,
    Soil_Type20,
    Soil_Type21,
    Soil_Type22,
    Soil_Type23,
    Soil_Type24,
    Soil_Type25,
    Soil_Type26,
    Soil_Type27,
    Soil_Type28,
    Soil_Type29,
    Soil_Type30,
    Soil_Type31,
    Soil_Type32,
    Soil_Type33,
    Soil_Type34,
    Soil_Type36,
    Soil_Type37,
```

```
    Soil_Type38,
    Soil_Type39,
    Soil_Type40,
    predict_cover_type( Elevation,
    Aspect,
    Slope,
    Horizontal_distance_to_hydrology,
    Vertical_distance_to_hydrology,
    Horizontal_distance_to_roadways,
    HIllshade_9am,
    Hillshade_noon,
    Hillshade_3pm ,
    Horizontal_Distance_To_Fire_Points,
    Wilderness_Area1,
    Wilderness_Area2,
    Wilderness_Area3,
    Wilderness_Area4,
    soil_type1,
    Soil_Type2,
    Soil_Type3,
    Soil_Type4,
    Soil_Type5,
    Soil_Type6,
    Soil_Type7,
    Soil_Type8,
    Soil_Type9,
    Soil_Type10,
    Soil_Type11,
    Soil_Type12,
    Soil_Type13,
    Soil_Type14,
    Soil_Type15,
    Soil_Type16,
    Soil_Type17,
    Soil_Type18,
    Soil_Type19,
    Soil_Type20,
    Soil_Type21,
    Soil_Type22,
    Soil_Type23,
    Soil_Type24,
    Soil_Type25,
    Soil_Type26,
    Soil_Type27,
```

```
   Soil_Type28,
   Soil_Type29,
   Soil_Type30,
   Soil_Type31,
   Soil_Type32,
   Soil_Type33,
   Soil_Type34,
   Soil_Type36,
   Soil_Type37,
   Soil_Type38,
   Soil_Type39,
   Soil_Type40) AS predicted_cover_type

FROM public.covertype_test
WHERE wilderness_area2 = 1)
t1
GROUP BY 1;
```

L'output del comando precedente dovrebbe essere simile all'esempio seguente. Questo risultato indica che il modello ha previsto che la maggior parte delle coperture è di tipo 1 e sono presenti coperture di tipo 2 e 7.

```
+----------------------+-------+
| predicted_cover_type | count |
+----------------------+-------+
|                    2 |   564 |
|                    7 |    97 |
|                    1 |  2309 |
+----------------------+-------+
```

3. La seguente query mostra il tipo di copertura più comune in una singola area naturale. La query visualizza la quantità del tipo di copertura specifico e l'area naturale associata al tipo di copertura.

```
SELECT t1. predicted_cover_type, COUNT(*), wilderness_area
FROM
(
SELECT
   Elevation,
   Aspect,
   Slope,
   Horizontal_distance_to_hydrology,
   Vertical_distance_to_hydrology,
```

```
    Horizontal_distance_to_roadways,
    HIllshade_9am,
    Hillshade_noon,
    Hillshade_3pm ,
    Horizontal_Distance_To_Fire_Points,
    Wilderness_Area1,
    Wilderness_Area2,
    Wilderness_Area3,
    Wilderness_Area4,
    soil_type1,
    Soil_Type2,
    Soil_Type3,
    Soil_Type4,
    Soil_Type5,
    Soil_Type6,
    Soil_Type7,
    Soil_Type8,
    Soil_Type9,
    Soil_Type10 ,
    Soil_Type11,
    Soil_Type12 ,
    Soil_Type13 ,
    Soil_Type14,
    Soil_Type15,
    Soil_Type16,
    Soil_Type17,
    Soil_Type18,
    Soil_Type19,
    Soil_Type20,
    Soil_Type21,
    Soil_Type22,
    Soil_Type23,
    Soil_Type24,
    Soil_Type25,
    Soil_Type26,
    Soil_Type27,
    Soil_Type28,
    Soil_Type29,
    Soil_Type30,
    Soil_Type31,
    Soil_Type32,
    Soil_Type33,
    Soil_Type34,
    Soil_Type36,
```

```
    Soil_Type37,
    Soil_Type38,
    Soil_Type39,
    Soil_Type40,
    predict_cover_type( Elevation,
    Aspect,
    Slope,
    Horizontal_distance_to_hydrology,
    Vertical_distance_to_hydrology,
    Horizontal_distance_to_roadways,
    HIllshade_9am,
    Hillshade_noon,
    Hillshade_3pm ,
    Horizontal_Distance_To_Fire_Points,
    Wilderness_Area1,
    Wilderness_Area2,
    Wilderness_Area3,
    Wilderness_Area4,
    soil_type1,
    Soil_Type2,
    Soil_Type3,
    Soil_Type4,
    Soil_Type5,
    Soil_Type6,
    Soil_Type7,
    Soil_Type8,
    Soil_Type9,
    Soil_Type10,
    Soil_Type11,
    Soil_Type12,
    Soil_Type13,
    Soil_Type14,
    Soil_Type15,
    Soil_Type16,
    Soil_Type17,
    Soil_Type18,
    Soil_Type19,
    Soil_Type20,
    Soil_Type21,
    Soil_Type22,
    Soil_Type23,
    Soil_Type24,
    Soil_Type25,
    Soil_Type26,
```

```
   Soil_Type27,
   Soil_Type28,
   Soil_Type29,
   Soil_Type30,
   Soil_Type31,
   Soil_Type32,
   Soil_Type33,
   Soil_Type34,
   Soil_Type36,
   Soil_Type37,
   Soil_Type38,
   Soil_Type39,
   Soil_Type40) AS predicted_cover_type,
   CASE WHEN Wilderness_Area1 = 1 THEN 1
        WHEN Wilderness_Area2 = 1 THEN 2
        WHEN Wilderness_Area3 = 1 THEN 3
        WHEN Wilderness_Area4 = 1 THEN 4
        ELSE 0
   END AS wilderness_area

FROM public.covertype_test)
t1
GROUP BY 1, 3
ORDER BY 2 DESC
LIMIT 1;
```

L'output del comando precedente dovrebbe essere simile all'esempio seguente.

```
+----------------------+-------+-----------------+
| predicted_cover_type | count | wilderness_area |
+----------------------+-------+-----------------+
|                    2 | 15738 |               1 |
+----------------------+-------+-----------------+
```

Argomenti correlati

Per ulteriori informazioni su Amazon Redshift ML, fare riferimento ai seguenti collegamenti:

- Costi per l'utilizzo di Amazon Redshift ML
- Operazione CREATE MODEL
- Funzione EXPLAIN_MODEL

Per ulteriori informazioni sul machine learning, consulta la documentazione seguente:

- Panoramica del machine learning
- Machine learning per principianti ed esperti
- Che cos'è l'equità e la spiegabilità dei modelli per le previsioni di Machine Learning?

Ottimizzazione delle prestazioni delle query

Amazon Redshift utilizza query basate sul linguaggio di query strutturato (SQL) per interagire con i dati e gli oggetti nel sistema. Il Data Manipulation Language (DML, linguaggio di manipolazione dei dati) è il sottoinsieme dell'SQL, utilizzato per visualizzare, aggiungere, modificare e cancellare dati. Il Data Definition Language (DDL, linguaggio di definizione dei dati) è il sottoinsieme dell'SQL, utilizzato per visualizzare, aggiungere, modificare e cancellare gli oggetti del database come tabelle e visualizzazioni.

Quando il tuo sistema viene configurato, lavorerai principalmente con il DML, in particolare con il comando SELECT per il recupero e la visualizzazione dei dati. Per scrivere query di recupero dati efficaci in Amazon Redshift, sarà necessario acquisire familiarità con l'istruzione SELECT e applicare i suggerimenti mostrati in Best practice di Amazon Redshift per la progettazione di tabelle per aumentare al massimo l'efficienza delle query.

Per comprendere il modo in cui Amazon Redshift elabora le query, consultare le sezioni Elaborazione query e Analisi e miglioramento delle query. Quindi potrai applicare queste informazioni in combinazioni con gli strumenti di diagnostica, al fine di identificare ed eliminare i problemi relativi alle prestazioni delle query.

Per identificare e risolvere alcuni dei problemi più comuni e importanti che potrebbero verificarsi con le query di Amazon Redshift, utilizzare la sezione Risoluzione dei problemi delle query.

Argomenti

Elaborazione query

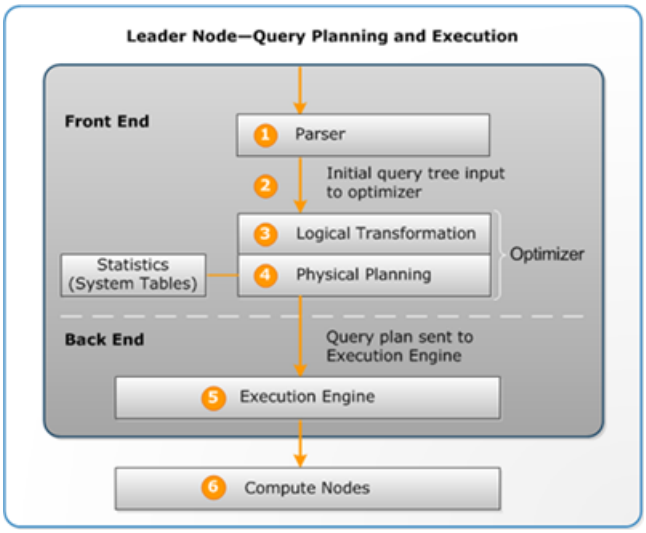
Amazon Redshift instrada una query SQL inviata attraverso il parser e l'ottimizzatore in modo da sviluppare un piano di query. Quindi il motore di esecuzione traduce il piano di query in un codice che invia ai nodi di calcolo per l'esecuzione.

Argomenti

- Revisione delle fasi del piano di query
- Fattori che influenzano le prestazioni della query

Pianificazione di query e flusso di lavoro di esecuzione

La seguente illustrazione fornisce una visualizzazione di alto livello della pianificazione della query e dell'esecuzione del flusso di lavoro.



La pianificazione della query e il flusso di lavoro dell'esecuzione seguono queste fasi:

1. Il nodo principale riceve la query e analizza l'SQL.
2. Il parser produce un struttura ad albero di query iniziale, ovvero una rappresentazione logica della query originale. Amazon Redshift quindi inserisce questa struttura ad albero di query nell'ottimizzatore di query.
3. L'ottimizzatore valuta e, se necessario, riscrive la query per massimizzare la sua efficienza. A volte il risultato di questo processo è la creazione di query multiple correlate che ne rimpiazzano una singola.
4. L'ottimizzatore genera un piano di query (o più di uno, se la fase precedente ha prodotto più query) per l'esecuzione con le prestazioni migliori. Il piano di query specifica le opzioni di esecuzione come il tipo di combinazione, l'ordine di combinazione, le opzioni di aggregazione e i requisiti di distribuzione dei dati.

   Puoi utilizzare il comando EXPLAIN per visualizzare il piano di query. Il piano di query è uno strumento fondamentale per l'analisi e l'ottimizzazione di query complesse. Per ulteriori informazioni, consulta Piano di query.

5. Il motore di esecuzione traduce il piano di query in fasi, segmenti e flussi:

   Fase

   Ogni fase è un'operazione individuale necessaria durante l'esecuzione di query. Le fasi possono essere combinate per permettere ai nodi di calcolo di eseguire una query, una combinazione o un'altra operazione del database.

   Segment

   Una combinazione di più fasi che può essere eseguita tramite un singolo processo; è inoltre l'unità di elaborazione più piccola eseguibile tramite una sezione di nodi di calcolo. Una sezione è l'unità di elaborazione parallela in Amazon Redshift. I segmenti in un flusso eseguiti in parallelo.

   Flusso

   Un insieme di segmenti che possono essere suddivisi sulle sezioni dei nodi di calcolo disponibili.

   Il motore di esecuzione genera un codice compilato basato su fasi, segmenti e flussi. I codici compilati vengono eseguiti più velocemente rispetto ai codici interpretati e utilizzano una minore capacità di elaborazione. Questo codice compilato viene poi trasmesso ai nodi di calcolo.

   > **Note**
   >
   > Quando esegui il benchmarking delle tue query, dovresti sempre confrontare i tempi della seconda esecuzione di una query, perché il tempo della prima esecuzione include i costi della compilazione del codice. Per ulteriori informazioni, consulta Fattori che influenzano le prestazioni della query.

6. Le sezioni di nodi computati eseguono i segmenti della query in parallelo. Essendo parte di questo processo, Amazon Redshift sfrutta la comunicazione di rete ottimizzata, la memoria e la gestione del disco per inviare risultati intermedi da una fase del piano di query alla successiva. Questo aiuta anche a velocizzare l'esecuzione delle query.

Le fasi 5 e 6 si verificano una sola volta per ogni flusso. Il motore crea i segmenti eseguibili per un flusso e li invia ai nodi di calcolo. Quando i segmenti di quel flusso sono completi, il motore genera i segmenti del flusso successivo. In questo modo, il motore riesce ad analizzare cosa è successo

nel flusso precedente (ad esempio, se le operazioni erano basate su disco), al fine di influenzare la generazione di segmenti nel prossimo flusso.

Quando i nodi di calcolo vengono terminati, restituiscono i risultati della query al nodo principale per eseguire l'elaborazione finale. Il nodo principale unisce i dati in un unico insieme di risultati e gestisce qualunque ordinamento o aggregazione di cui si necessiti. Il nodo principale infine restituisce i risultati al client.

> **Note**
>
> I nodi di calcolo, se necessario, possono restituire dei dati al nodo principale durante l'esecuzione della query. Ad esempio, se disponi di una sottoquery con una clausola LIMIT, il limite viene applicato sul nodo principale prima che i dati vengano distribuiti nuovamente all'interno del cluster per elaborazioni future.

Piano di query

Per ottenere informazioni sulle operazioni individuali necessarie per eseguire una query, puoi utilizzare il piano di query. Prima di lavorare con il piano di query, consigliamo di capire in che modo Amazon Redshift gestisce l'elaborazione delle query e crea piani di query. Per ulteriori informazioni, consultare Pianificazione di query e flusso di lavoro di esecuzione.

Per creare un piano di query, esegui il comando EXPLAIN seguito dal testo di query corrente. Il piano di query rende le seguenti informazioni:

- Quali operazioni eseguirà il motore di esecuzione, leggendo i risultati dal basso verso l'alto.
- Che tipo di fase esegue ogni operazione.
- Quali tabelle e colonne vengono utilizzate in ogni operazione.
- Quanti dati vengono eseguiti in ogni operazione, in termini di numero delle file e di larghezza dei dati in byte.
- Il costo relativo dell'operazione. Il costo è una misura che confronta i tempi di esecuzione relativi delle fasi all'interno di un piano. Il costo non fornisce alcuna informazioni precisa in merito all'attuale consumo di memoria o ai tempi di esecuzione; non fornisce inoltre un confronto significativo tra i piani di esecuzione. Non ti fornisce un'indicazione di quali operazioni stanno consumando più risorse in una query.

Il comando EXPLAIN non esegue effettivamente la query. Mostra solamente il piano che verrà eseguito da Amazon Redshift se la query viene eseguita nelle condizioni operative attuali. Se modifichi lo schema o i dati di una tabella ed esegui nuovamente il comando ANALYZE per aggiornare i metadati statistici, il piano di query potrebbe essere differente.

L'output del piano di query che genera il comando EXPLAIN, è una visualizzazione semplificata e di alto livello dell'esecuzione della query. Non illustra i dettagli dell'elaborazione di query parallela. Per visualizzare le informazioni dettagliate, devi eseguire la stessa query, quindi ottenere le informazioni di riepilogo della query dalle visualizzazioni SVL_QUERY_SUMMARY o SVL_QUERY_REPORT. Per ulteriori informazioni sull'utilizzo di queste visualizzazioni, consultare Analisi del riepilogo della query.

Il seguente esempio illustra l'output EXPLAIN per una semplice query GROUP BY sulla tabella EVENT:

```
explain select eventname, count(*) from event group by eventname;

                            QUERY PLAN
------------------------------------------------------------------
XN HashAggregate  (cost=131.97..133.41 rows=576 width=17)
  ->  XN Seq Scan on event  (cost=0.00..87.98 rows=8798 width=17)
```

EXPLAIN restituisce i seguenti parametri per ogni operazione:

Costo

Un valore relativo utile per il confronto delle operazioni all'interno di un piano. Il costo è costituito da due valori decimali separati da due punti, ad esempio `cost=131.97..133.41`. Il primo valore, in questo caso 131.97, fornisce il costo relativo della restituzione della prima riga per questa operazione. Il secondo valore, in questo caso 133.41, fornisce il costo relativo del completamento dell'operazione. I costi del piano di query sono cumulativi man mano che si legge il piano, quindi il HashAggregate costo in questo esempio (131,97.. 133.41) include il costo del Seq Scan sottostante (0,00.. 87,98).

Righe

Il numero di righe previsto da restituire. In questo esempio, si prevede che la scansione restituisca 8798 righe. L' HashAggregate operatore da solo dovrebbe restituire 576 righe (dopo che i nomi di eventi duplicati sono stati eliminati dal set di risultati).

> **Note**
>
> La stima delle righe si basa sulle statistiche disponibili generate dal comando ANALYZE. Se il comando ANALYZE non è stato eseguito di recente, la stima sarà meno affidabile.

Larghezza

La larghezza stimata della riga media, misurata in byte. In questo esempio, si prevede che la riga media sia larga 17 byte.

Operatori EXPLAIN

Questa sezione descrive brevemente gli operatori che visualizzi più spesso nell'output EXPLAIN. Per un elenco completo degli operatori, consultare EXPLAIN nella sezione relativa ai comandi SQL.

Operatore scansione sequenziale

L'operatore scansione sequenziale (Seq Scan) indica una scansione della tabella. Seq Scan scansiona ogni colonna della tabella in maniera sequenziale dall'inizio alla fine, valutando le limitazioni della query (nella clausola WHERE) per ogni riga.

Operatori di combinazione

Amazon Redshift seleziona gli operatori join basati sul progetto fisico delle tabelle combinate, la posizione dei dati necessari per il join e i requisiti specifici della stessa query.

- Loop nidificato

  La combinazione meno ottimale è un loop nidificato, utilizzato principalmente per le combinazioni incrociate (prodotti cartesiano) e per alcune combinazioni di disuguaglianza.
- Operatori hash join e hash

  Generalmente più veloci di una combinazione loop nidificata, un operatore hash join e hash viene utilizzato per le inner join e per le outer join destra e sinistra. Questi operatori vengono utilizzati nel caso cui le tabelle combinate in cui si trovano le colonne di combinazione non siano chiavi di distribuzione e chiavi di ordinamento. L'operatore hash crea la tabella hash per la tabella interna nella combinazione; l'operatore hash join legge la tabella esterna, esegue la colonna di combinazione e trova corrispondenze nella tabella interna hash.

- Merge join

  Un merge join, generalmente la combinazione più veloce, viene utilizzato per le inner join e le outer join. Merge join non si utilizza per le full join. Questo operatore viene utilizzato nel caso cui le tabelle combinate in cui si trovano le colonne di combinazione siano chiavi di distribuzione e chiavi di ordinamento; inoltre viene utilizzato quando meno del 20 percento delle tabelle combinate non viene ordinato. Legge due tabelle ordinate e trova le righe corrispondenti. Per visualizzare la percentuale delle righe non ordinate, eseguire una query della tabella di sistema SVV_TABLE_INFO.
- Spatial Join

  In genere un join rapido basato sulla prossimità dei dati spaziali, utilizzati per i tipi di dati `GEOMETRY` e `GEOGRAPHY`.

Operatori di aggregazione

Il piano di query utilizza i seguenti operatori nelle query che implicano le funzioni aggregate e le operazioni GROUP BY.

- Aggregazione

  L'operatore per le funzioni aggregate scalari, come AVG e SUM.
- HashAggregate

  L'operatore per le funzioni aggregate raggruppate in modo non ordinato.
- GroupAggregate

  L'operatore per le funzioni aggregate raggruppate in modo ordinato.

Operatori di ordinamento

Il piano di query utilizza i seguenti operatori quando le query devono ordinare o unire gli insiemi dei risultati.

- Ordina

  Valuta le clausole ORDER BY e altre operazioni di ordinamento, come gli ordinamenti necessari per le query e le combinazioni UNION, le query SELECT DISTINCT e le funzioni finestra.
- Unione

Produce risultati ordinati finali a seconda dei risultati ordinati intermedi che derivano dalle operazioni parallele.

Operatori UNION, INTERSECT ed EXCEPT

Il piano di query utilizza i seguenti operatori per le query che implicano le funzioni aggregate e le operazioni dell'insieme con UNION, INTERSECT e EXCEPT.

- Sottoquery

  Utilizzate per eseguire le query UNION.
- Hash Intersect Distinct

  Utilizzato per eseguire query INTERSECT .
- SetOp Eccetto

  Utilizzato per eseguire le query EXCEPT (o MINUS).

Altri operatori

I seguenti operatori appaiono frequentemente anche nell'output EXPLAIN per le query di routine.

- Unique

  Elimina i duplicati per le query SELECT DISTINCT e per le query UNION.
- Limite

  Elabora la clausola LIMIT.
- Window

  Esegue le funzioni finestra.
- Risultato

  Esegue le funzioni scalati che non implicano alcun accesso di tabella.
- Subplan

  Utilizzato per determinate sottoquery.
- Rete

Invia risultati intermedi al nodo principale per elaborazioni future.

- Materialize

  Salva le righe per l'ingresso delle combinazioni del loop nidificato e alcuni merge join.

Combinazioni su EXPLAIN

L'ottimizzatore di query utilizza diversi tipi di combinazioni per recuperare i dati della tabella, a seconda della struttura della query e delle tabelle sottostanti. L'output EXPLAIN fa riferimento al tipo di combinazione, alla tabella utilizzata e al modo in cui i dati della tabella vengono distribuiti all'interno del cluster, al fine di descrivere in che modo viene elaborata la query.

Esempi di tipo di join

I seguenti esempi mostrano i diversi tipi di combinazione che può utilizzare l'ottimizzatore di query. Il tipo di combinazione utilizzato nel piano della query dipende dal progetto fisico delle tabelle coinvolte.

Esempio: eseguire un hash join di due tabelle

La seguente query combina EVENT e CATEGORY sulla colonna CATID. CATID è la chiave di distribuzione e di ordinamento per CATEGORY, ma non per EVENT. Un hash join viene eseguito con EVENT come tabella esterna e CATEGORY come tabella interna. Dato che CATEGORY è la tabella più piccola, il pianificatore trasmette una copia della stessa ai nodi di calcolo durante l'elaborazione della query, mediante l'uso di DS_BCAST_INNER. Il costo di combinazione in questo esempio rappresenta la maggior parte del costo cumulativo del piano.

```
explain select * from category, event where category.catid=event.catid;

                              QUERY PLAN
------------------------------------------------------------------------
 XN Hash Join DS_BCAST_INNER  (cost=0.14..6600286.07 rows=8798 width=84)
   Hash Cond: ("outer".catid = "inner".catid)
   ->  XN Seq Scan on event  (cost=0.00..87.98 rows=8798 width=35)
   ->  XN Hash  (cost=0.11..0.11 rows=11 width=49)
         ->  XN Seq Scan on category  (cost=0.00..0.11 rows=11 width=49)
```

> **Note**
>
> I rientri allineati degli operatori nell'output EXPLAIN, indicano a volte che quelle operazioni non dipendono l'una dall'altra e che possono iniziare in parallelo. Nell'esempio precedente, nonostante la scansione sulla tabella EVENT e le operazioni hash siano allineate, la scansione EVENT deve attendere il completamento totale dell'operazione hash.

Esempio: eseguire un merge join di due tabelle

La seguente query utilizza anche SELECT *, ma combina SALES e LISTING sulla colonna LISTID, dove LISTID è stato impostato come chiave di distribuzione e come chiave di ordinamento per entrambe le tabelle. Viene scelto un merge join e per la combinazione non è necessaria alcuna ridistribuzione dei dati (DS_DIST_NONE).

```
explain select * from sales, listing where sales.listid = listing.listid;
QUERY PLAN
------------------------------------------------------------------------------
XN Merge Join DS_DIST_NONE  (cost=0.00..6285.93 rows=172456 width=97)
  Merge Cond: ("outer".listid = "inner".listid)
  ->  XN Seq Scan on listing  (cost=0.00..1924.97 rows=192497 width=44)
  ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=53)
```

L'esempio seguente dimostra i diversi tipi di combinazione all'interno della stessa query. Come nell'esempio precedente, SALES e LISTING vengono combinati mediante l'operatore merge join, ma la terza tabella EVENT deve essere combinata con i risultati dell'operatore merge join. Ancora una volta, l'operatore merge join incorre in un costo di trasmissione.

```
explain select * from sales, listing, event
where sales.listid = listing.listid and sales.eventid = event.eventid;
                                  QUERY PLAN
------------------------------------------------------------------------------
XN Hash Join DS_BCAST_INNER  (cost=109.98..3871130276.17 rows=172456 width=132)
  Hash Cond: ("outer".eventid = "inner".eventid)
  ->  XN Merge Join DS_DIST_NONE  (cost=0.00..6285.93 rows=172456 width=97)
        Merge Cond: ("outer".listid = "inner".listid)
        ->  XN Seq Scan on listing  (cost=0.00..1924.97 rows=192497 width=44)
        ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=53)
  ->  XN Hash  (cost=87.98..87.98 rows=8798 width=35)
        ->  XN Seq Scan on event  (cost=0.00..87.98 rows=8798 width=35)
```

Esempi: Join, Aggregate e Sort

La seguente query esegue un hash join delle tabelle SALES e EVENT, seguito da operazioni di distribuzione e aggregazione per verificare la funzione SUM raggruppata e la clausola ORDER BY. L'operatore Sort iniziale viene eseguito in parallelo sui nodi di calcolo. Quindi l'operatore Network invia i risultati al nodo principale, dove l'operatore Merge produce i risultati finali ordinati.

```
explain select eventname, sum(pricepaid) from sales, event
where sales.eventid=event.eventid group by eventname
order by 2 desc;
                                          QUERY PLAN
------------------------------------------------------------------------------
 XN Merge  (cost=1002815366604.92..1002815366606.36 rows=576 width=27)
  Merge Key: sum(sales.pricepaid)
  ->  XN Network  (cost=1002815366604.92..1002815366606.36 rows=576 width=27)
        Send to leader
        ->  XN Sort  (cost=1002815366604.92..1002815366606.36 rows=576 width=27)
              Sort Key: sum(sales.pricepaid)
              ->  XN HashAggregate  (cost=2815366577.07..2815366578.51 rows=576
 width=27)
                    ->  XN Hash Join DS_BCAST_INNER  (cost=109.98..2815365714.80
 rows=172456 width=27)
                          Hash Cond: ("outer".eventid = "inner".eventid)
                          ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456
 width=14)
                          ->  XN Hash  (cost=87.98..87.98 rows=8798 width=21)
                                ->  XN Seq Scan on event  (cost=0.00..87.98 rows=8798
 width=21)
```

Ridistribuzione dei dati

L'output EXPLAIN per le combinazioni, inoltre, specifica un metodo per come verranno spostati i dati all'interno di un cluster, così che la combinazione sia più semplice. Questo movimento di dati può essere sia una trasmissione che una ridistribuzione. In una trasmissione, i valori dei dati da un lato della combinazione vengono copiati da ogni nodo di calcolo su ogni altro nodo di calcolo, così che ognuno di essi finisca con una copia completa dei dati. In una ridistribuzione, la partecipazione dei valori dei dati viene inviata dalla loro sezione attuale alla nuova sezione (possibilmente su un nodo differente). Solitamente i dati vengono ridistribuiti per far corrispondere la chiave di distribuzione dell'altra tabella che ha partecipato alla combinazione, nel caso in cui la chiave di distribuzione sia una delle colonne di combinazione. Se nessuna delle tabelle dispone di chiavi di distribuzione su

una delle colonne di combinazione, entrambe le tabelle vengono distribuite o la tabella interna viene trasmessa a ogni nodo.

L'output EXPLAIN fa riferimento anche alle tabelle esterne e interne. La tabella interna viene scansionata per prima e appare più vicina al fondo del piano di query. La tabella interna è la tabella in cui si trovano le corrispondenza. Questa si trova solitamente in memoria, di solito è la tabella di origine per l'hashing e, probabilmente, è la tabella più piccola delle due combinate. La tabella esterna è la sorgente delle righe da combinare con la tabella interna. Viene letta solitamente dal disco. L'ottimizzatore di query sceglie la tabella interne e quella esterna a seconda delle statistiche del database, che provengono dall'esecuzione più recente del comando ANALYZE. L'ordine delle tabelle nella clausola FROM di una query non determina quale sia la tabella esterne né quella interna.

Utilizza i seguenti attributi nei piani di query, per identificare come verranno spostati i dati per facilitare una query:

- DS_BCAST_INNER

  Una copia di tutta la tabella interna viene trasmessa a tutti i nodi di calcolo.
- DS_DIST_ALL_NONE

  Non è necessaria alcuna ridistribuzione, perché la tabella interna è già stata distribuita su ogni nodo mediante l'uso di DISTSTYLE ALL.
- DS_DIST_NONE

  Nessuna tabella viene ridistribuita. È possibile disporre di combinazioni collocate, nel caso in cui le sezioni corrispondenti vengano combinate senza lo spostamento dei dati tra i nodi.
- DS_DIST_INNER

  La tabella interna viene ridistribuita.
- DS_DIST_OUTER

  La tabella esterna viene ridistribuita.
- DS_DIST_ALL_INNER

  Tutta la tabella interna viene ridistribuita in una singola sezione, perché la tabella esterna utilizza DISTSTYLE ALL.
- DS_DIST_BOTH

  Entrambe le tabelle vengono ridistribuite.

Revisione delle fasi del piano di query

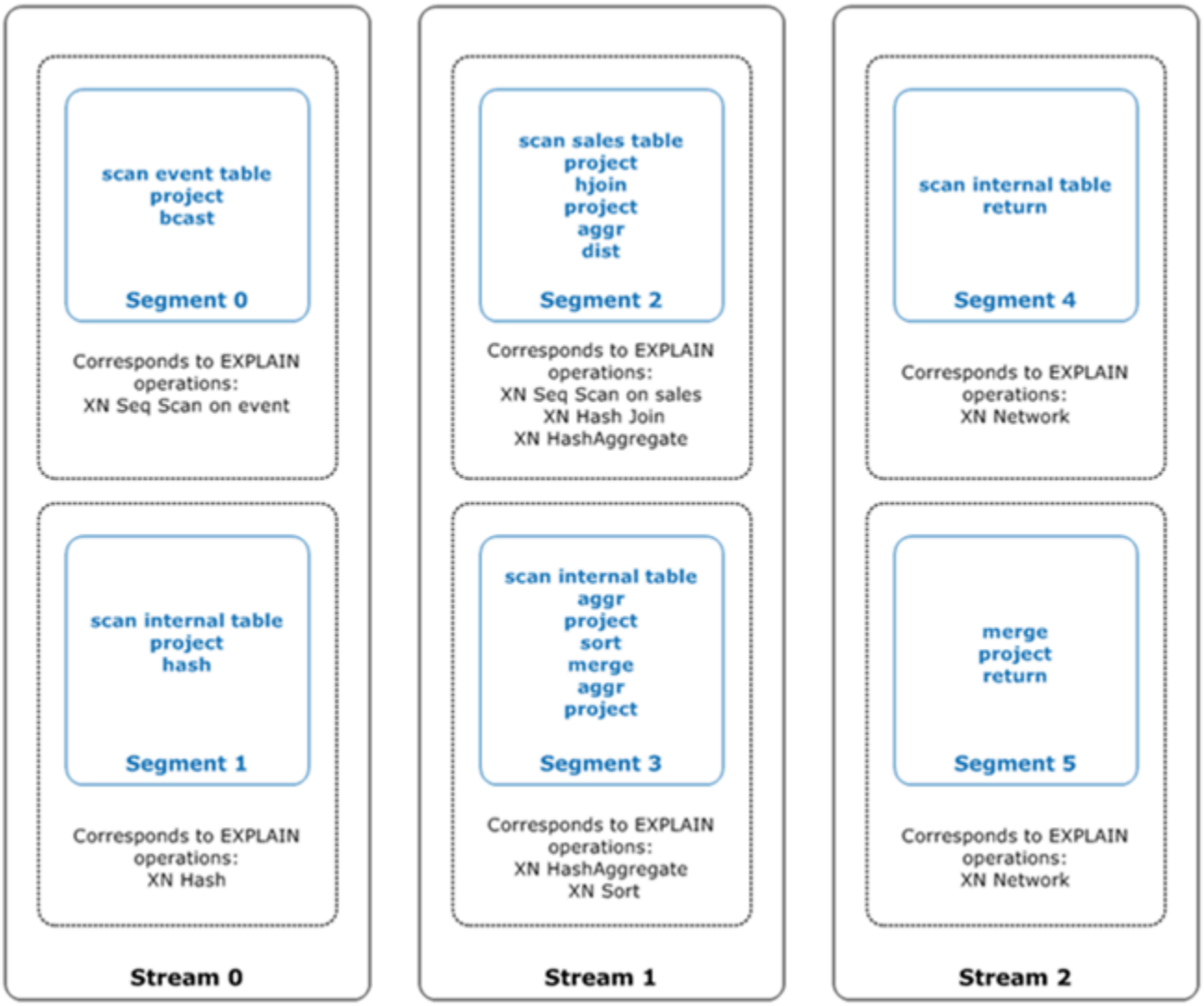
Puoi visualizzare le fasi in un piano di query eseguendo il comando EXPLAIN. L'esempio seguente mostra una query SQL e spiega l'output. Leggendo il piano di query dal basso verso l'alto, puoi visualizzare tutte le operazioni logiche necessarie per eseguire la query. Per ulteriori informazioni, consultare Piano di query.

```
explain
select eventname, sum(pricepaid) from sales, event
where sales.eventid = event.eventid
group by eventname
order by 2 desc;
```

```
XN Merge  (cost=1002815366604.92..1002815366606.36 rows=576 width=27)
  Merge Key: sum(sales.pricepaid)
  ->  XN Network  (cost=1002815366604.92..1002815366606.36 rows=576 width=27)
        Send to leader
        ->  XN Sort  (cost=1002815366604.92..1002815366606.36 rows=576 width=27)
              Sort Key: sum(sales.pricepaid)
              ->  XN HashAggregate  (cost=2815366577.07..2815366578.51 rows=576
 width=27)
                    ->  XN Hash Join DS_BCAST_INNER  (cost=109.98..2815365714.80
 rows=172456 width=27)
                          Hash Cond: ("outer".eventid = "inner".eventid)
                          ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456
 width=14)
                          ->  XN Hash  (cost=87.98..87.98 rows=8798 width=21)
                                ->  XN Seq Scan on event  (cost=0.00..87.98 rows=8798
 width=21)
```

Come parte della generazione di un piano di query, l'ottimizzatore query suddivide il piano in flussi, segmenti e passaggi. L'ottimizzatore query interrompe il piano per prepararsi alla distribuzione dei dati e del carico di lavoro di query ai nodi di calcolo. Per ulteriori informazioni sui segmenti e sulle fasi, consultare Pianificazione di query e flusso di lavoro di esecuzione.

L'illustrazione seguente mostra la query precedente e il piano di query associato. Viene visualizzato il modo in cui le operazioni di query coinvolte eseguono la mappatura dei passaggi utilizzati da Amazon Redshift per generare un codice compilato per le sezioni del nodo di calcolo. Ogni operazione del piano di query viene mappato per più fasi all'interno dei segmenti, talvolta per più segmenti all'interno dei flussi.



In questa illustrazione, query optimizer esegue il piano di query in questo modo:

1. In `Stream 0`, la query esegue `Segment 0` con un'operazione di scansione sequenziale per eseguire la scansione della tabella `events`. La query continua a `Segment 1` con un'operazione hash per creare la tabella hash per la tabella interna nel join.
2. In `Stream 1`, la query esegue `Segment 2` con un'operazione di scansione sequenziale per eseguire la scansione della tabella `sales`. Continua con `Segment 2` con un hash join per unire le tabelle in cui le colonne join non sono né chiavi di distribuzione né chiavi di ordinamento. Continua di nuovo con `Segment 2` con un aggregato hash per aggregare i risultati. Quindi la query viene eseguita con `Segment 3`, un'operazione di aggregazione hash per eseguire funzioni aggregate raggruppate non ordinate, e un'operazione di ordinamento per valutare la clausola ORDER BY e altre operazioni di ordinamento.

3. In `Stream 2`, la query esegue un'operazione di rete in `Segment 4` e `Segment 5` per inviare risultati intermedi al nodo leader per ulteriori elaborazioni.

L'ultimo segmento di una query restituisce i dati. Se il set restituito è aggregato o ordinato, i nodi di calcolo inviano ciascuno la parte del risultato intermedio al nodo leader. Il nodo direttivo quindi unisce i dati in modo che il risultato finale possa essere inviato nuovamente al client richiedente.

Per ulteriori informazioni sugli operatori EXPLAIN, vedere EXPLAIN.

Fattori che influenzano le prestazioni della query

Vari fattori possono avere delle ripercussioni sulle prestazioni delle query. I seguenti aspetti dei tuoi dati, dei cluster e delle operazioni del database giocano tutti un ruolo fondamentale su quanto velocemente verranno elaborate le query.

- Numero di nodi, processori o sezioni: un nodo di calcolo è diviso in sezioni. Più nodi significano più processori e più sezioni. Queste consentono alle tue query di essere elaborate più velocemente attraverso l'esecuzione in contemporanea delle porzioni della query all'interno delle sezioni. Tuttavia, più nodi significano anche una spesa maggiore, quindi avrai bisogno di trovare l'equilibrio tra costi e prestazioni proprie del tuo sistema. Per ulteriori informazioni sull'architettura dei cluster Amazon Redshift, consultare Architettura del sistema di data warehouse.
- Node types (Tipi di nodi): un cluster Amazon Redshift può utilizzare uno dei diversi tipi di nodi. Ogni nodo presenta differenti dimensioni e limiti, per aiutarti a scalare il cluster in maniera appropriata. La dimensione del nodo determina la capacità di storage, la memoria, la CPU e il prezzo di ogni nodo nel cluster. Per ulteriori informazioni sui tipi di nodo, consulta Panoramica dei cluster di Amazon Redshift nella Guida alla gestione di Amazon Redshift.
- Distribuzione dei dati: Amazon Redshift archivia i dati della tabella sui nodi di calcolo in base allo stile della distribuzione della tabella. Quando si esegue una query, l'ottimizzatore di query ridistribuisce i dati ai nodi di calcolo per eseguire qualsiasi combinazione e aggregazione, in base alle necessità. La scelta del corretto stile di distribuzione per una tabella, aiuta a minimizzare l'impatto della fasi di ridistribuzione posizionando i dati dove servono prima dell'esecuzione delle combinazioni. Per ulteriori informazioni, consultare Utilizzo degli stili di distribuzione dati.
- Ordinamento dei dati: Amazon Redshift archivia in modo ordinato i dati della tabella sul disco in base alle chiavi di ordinamento della tabella. L'ottimizzatore e l'elaboratore di query utilizzano le informazioni sulla posizione dei dati per ridurre il numero dei blocchi da scansionare, con il conseguente miglioramento della velocità di query. Per ulteriori informazioni, consultare Utilizzo delle chiavi di ordinamento.

- Dimensioni set di dati: un volume di dati maggiore nel cluster può rallentare le prestazioni delle query perché è necessario eseguire la scansione e ridistribuire un numero maggiore di righe. Puoi ridurre questo effetto eseguendo il vacuum e archiviando i dati regolarmente, oltre che mediante l'uso di un predicato che limiti l'insieme dei dati della query.
- Operazioni simultanee: l'esecuzione contemporanea di più operazioni può avere ripercussioni sulle prestazioni della query. Ogni operazione impiega uno o più slot in una coda di query disponibile e utilizza la memoria relativa a questi slot. Se si stanno eseguendo altre operazioni, è possibile che non siano sufficienti gli slot della coda di query disponibili. In questo caso, la query dovrà attendere che si aprano gli slot prima che possa cominciare l'elaborazione. Per ulteriori informazioni sulla creazione e configurazione di code di query, consultare Implementazione della gestione del carico di lavoro.
- Struttura della query: il modo in cui viene scritta la query avrà delle ripercussioni sulle relative prestazioni. Per quanto possibile, scrivi le query affinché elaborino e restituiscano la minor quantità di dati possibile per le tue necessità. Per ulteriori informazioni, consultare Best practice di Amazon Redshift per la progettazione di query.
- Compilazione del codice: Amazon Redshift genera e compila il codice per ogni piano di esecuzione di query.

  Il codice compilato viene eseguito più velocemente perché elimina le spese di gestione derivanti dall'utilizzo di un interprete. Avrai sempre costi di gestione la prima volta che verrà generato e compilato il codice. Di conseguenza, le prestazioni di una query possono essere fuorvianti la prima volta che la esegui. I costi di gestione possono essere più evidenti quando esegui query occasionali (ad hoc). Eseguire la query una seconda volta per determinare le sue prestazioni tipiche. Amazon Redshift utilizza un servizio di compilazione serverless per ridimensionare le compilazioni di query oltre le risorse di calcolo di un cluster Amazon Redshift. I segmenti del codice compilato vengono memorizzati in locale nel cluster e in una cache virtuale illimitata. Questa cache persiste dopo il riavvio del cluster. Le successive esecuzioni della stessa query possono essere eseguite più rapidamente, in quanto è possibile saltare la fase di compilazione.

  La cache non è compatibile tra le versioni di Amazon Redshift, quindi la cache di compilazione viene svuotata e il codice viene ricompilato quando le query vengono eseguite dopo un aggiornamento della versione. Se le query hanno SLA rigorosi, consigliamo di eseguire in anticipo i segmenti di query che scansionano i dati dalle tabelle dei cluster. Ciò consente ad Amazon Redshift di memorizzare nella cache i dati della tabella di base, riducendo i tempi di pianificazione delle query dopo un aggiornamento della versione. Utilizzando un servizio di compilazione scalabile, Amazon Redshift è in grado di compilare codice in parallelo per fornire prestazioni

rapide e costanti. L'entità dell'accelerazione del carico di lavoro dipende dalla complessità e dalla simultaneità delle query.

Analisi e miglioramento delle query

Il recupero delle informazioni da un data warehouse di Amazon Redshift include l'esecuzione di query complesse su una quantità di dati molto vasta, operazione la cui elaborazione può impiegare molto tempo. Per assicurare un'elaborazione delle query più veloce possibile, esistono vari strumenti che puoi utilizzare per identificare possibili problemi di prestazioni.

Argomenti

Flusso di lavoro dell'analisi di query

Se una query sta impiegando più tempo del previsto, utilizza le seguenti fasi per rilevare e correggere il problema che potrebbe aver avuto ripercussioni negative sulle prestazioni della query. Se non sei sicuro di quali siano le query nel tuo sistema che possono trarre un vantaggio dall'ottimizzazione delle prestazioni, avvia l'esecuzione della query di diagnostica su Identificazione delle migliori query per l'ottimizzazione.

1. Assicurati che le tue tabelle siano progettate secondo le best practice. Per ulteriori informazioni, consulta Best practice di Amazon Redshift per la progettazione di tabelle.
2. Verifica se puoi eliminare o archiviare qualche dato non necessario nelle tue tabelle. Ad esempio, supponiamo che le tue query includano sempre i dati degli ultimi sei mesi, ma nelle tabelle disponi dei dati degli ultimi 18 mesi. In questo caso, puoi eliminare o archiviare i dati più vecchi per ridurre il numero di registri da dover scansionare e distribuire.
3. Esegui il comando VACUUM sulle tabelle nelle query per recuperare spazio e per ordinare nuovamente le righe. L'esecuzione del comando VACUUM è utile nel caso in cui la regione non

ordinata sia di grandi dimensioni e, inoltre, se la query utilizza la chiave di ordinamento in una combinazione o nel predicato.

4. Esegui il comando ANALYZE sulle tabelle nelle query per avere la certezza che le statistiche siano aggiornate. L'esecuzione del comando ANALYZE è utile nel caso in cui qualche tabella nella query sia stata modificata recentemente nelle sue dimensioni. Se l'esecuzione completa del comando ANALYZE impiega troppo tempo, esegui ANALYZE su una singola colonna per ridurre il tempo di elaborazione. Questo approccio aggiornerà comunque le statistiche della dimensione della tabella: quest'ultima infatti rappresenta un valore significativo nella pianificazione della query.
5. Assicurati che la tua query sia stata eseguita una volta per ogni tipo di client (a seconda del tipo di protocollo di connessione utilizzato dal client), così che la query venga compilata e memorizzata nella cache. Grazie a questo approccio, vengono velocizzate le esecuzioni successive di query. Per ulteriori informazioni, consultare Fattori che influenzano le prestazioni della query.
6. Verifica la tabella STL_ALERT_EVENT_LOG per rilevare e correggere possibili errori della tua query. Per ulteriori informazioni, consultare Revisione degli avvisi di query.
7. Esegui il comando EXPLAIN per ottenere il piano di query e utilizzarlo per ottimizzare la query. Per ulteriori informazioni, consultare Analisi del piano di query.
8. Utilizza le visualizzazioni SVL_QUERY_SUMMARY e SVL_QUERY_REPORT per ottenere le informazioni di riepilogo e utilizzarle per ottimizzare la query. Per ulteriori informazioni, consulta Analisi del riepilogo della query.

A volte, una query che dovrebbe essere eseguita velocemente viene forzata ad attendere, fino al completamento di un'altra query che ha un'esecuzione più lunga. In questo caso, potresti non aver nulla da migliorare nella query, ma puoi migliorare le prestazioni complessive del sistema, attraverso la creazione e l'utilizzo di code di query per differenti tipi di query. Per avere un'idea del tempo di attesa per la tua query, consultare Revisione dei tempi di attesa della coda per le query. Per ulteriori informazioni sulla configurazione di code di query, consultare Implementazione della gestione del carico di lavoro.

Revisione degli avvisi di query

Per utilizzare la tabella di sistema STL_ALERT_EVENT_LOG al fine di rilevare e correggere potenziali problemi relativi alle prestazioni della tua query, segui queste fasi:

1. Esegui la seguente operazione per determinare l'ID della tua query:

```
select query, elapsed, substring
```

```
from svl_qlog
order by query
desc limit 5;
```

Esamina il testo troncato della query nel campo `substring` per determinare quale valore `query` selezionare. Se hai eseguito la query più di una volta, utilizza il valore `query` a partire dalla riga con il valore `elapsed` più basso. Questa è la riga per la versione compilata. Se stai eseguendo molte query, puoi alzare il valore utilizzato dalla clausola LIMIT, che viene usata per accertarsi che la query sia inclusa.

2. Seleziona le righe da STL_ALERT_EVENT_LOG per la tua query:

```
Select * from stl_alert_event_log where query = MyQueryID;
```

userid	query	slice	segment	step	pid	xid	event	solution	event_time
100	32359	4	0	0	8780	71195	Very selective query filter:ratio=rows(2)/r	Review the choice of sort key to enable...	2015-02-10 17:40:50
100	32359	5	0	0	8781	71195	Very selective query filter:ratio=rows(2)/r	Review the choice of sort key to enable...	2015-02-10 17:40:50
100	109142	4	0	0	8780	302411	Very selective query filter:ratio=rows(2)/r	Review the choice of sort key to enable...	2015-02-24 20:32:28
100	109142	5	0	0	8781	302411	Very selective query filter:ratio=rows(2)/r	Review the choice of sort key to enable...	2015-02-24 20:32:28
100	109828	4	1	0	8746	304543	Very selective query filter:ratio=rows(3)/r	Review the choice of sort key to enable...	2015-02-24 23:27:52
100	109828	5	1	0	8747	304543	Very selective query filter:ratio=rows(3)/r	Review the choice of sort key to enable...	2015-02-24 23:27:52
100	109829	4	1	0	8760	304543	Very selective query filter:ratio=rows(3)/r	Review the choice of sort key to enable...	2015-02-24 23:28:01
100	109829	5	1	0	8761	304543	Very selective query filter:ratio=rows(3)/r	Review the choice of sort key to enable...	2015-02-24 23:28:01
100	113910	4	1	0	8774	316848	Very selective query filter:ratio=rows(3)/r	Review the choice of sort key to enable...	2015-02-25 17:14:58

3. Valuta i risultati della tua query. Utilizza la seguente tabella per individuare possibili soluzioni per ogni problema che hai rilevato.

> **Note**
>
> Non tutte le query disporranno di righe su STL_ALERT_EVENT_LOG, solo quelle per cui sono stati rilevati dei problemi.

Problema	Valore evento	Valore soluzione	Soluzione consigliata
Le statistiche delle tabelle nelle query sono mancanti o scadute.	Statistiche pianificatore query mancanti	Esegui comando ANALYZE	Per informazioni, consultare Statistiche della tabella mancanti o scadute.

Problema	Valore evento	Valore soluzione	Soluzione consigliata
C'è una combinazione di loop nidificati (la combinazione meno ottimale) nel piano di query.	Combinazione loop nidificata nel piano di query	Rivedi i predicati di combinazione per evitare prodotti cartesiani	Per informazioni, consultare Loop nidificato.
La scansione ha ignorato un numero piuttosto grande di righe contrassegnate come cancellate ma non svuotate, o righe che sono state inserite ma non completate.	Scansionato un grande numero di righe cancellate	Esegui il comando VACUUM per recuperare lo spazio cancellato	Per informazioni, consultare Righe fantasma o righe di cui non è stato eseguito il commit.
Più di 1.000.000 di righe sono state redistribuite per un hash join o un'aggregazione.	Ha distribuito un gran numero di righe sulla rete: RowCount le righe sono state distribuite per elaborare l'aggregazione	Rivedi la scelta della chiave di distribuzione per posizionare la combinazione o l'aggregazione	Per informazioni, consultare Distribuzione dei dati non ottimale.
Più di 1.000.000 di righe sono state trasmesse per un hash join.	Trasmesso un grande numero di righe all'interno della rete	Rivedi la scelta della chiave di distribuzione per posizionare la combinazione e per considerare l'utilizzo di tabelle distribuite	Per informazioni, consultare Distribuzione dei dati non ottimale.

Problema	Valore evento	Valore soluzione	Soluzione consigliata
Uno stile di redistribuzione DS_DIST_ALL_INNER è stato indicato nel piano di query, il che forza un'esecuzione seriale perché l'intera tabella interna è stata redistribuita in un solo nodo.	DS_DIST_ALL_INNER per un hash join nel piano di query	Rivedi la scelta della strategia di distribuzione per distribuire la tabella interna, piuttosto che quella esterna	Per informazioni, consultare Distribuzione dei dati non ottimale.

Analisi del piano di query

Prima di analizzare il piano di query, è necessario acquisire familiarità sulla sua lettura. Se non hai familiarità con la lettura di un piano di query, consigliamo di consultare Piano di query prima di continuare.

Esegui il comando EXPLAIN per ottenere un piano di query. Per analizzare i dati forniti dal piano di query, segui queste fasi:

1. Individua le fasi con il costo più alto. Concentrati sulla loro ottimizzazioni quando procederai con le fasi rimanenti.
2. Guarda i tipi di combinazione:
   - Nested Loop (Loop nidificato): combinazioni di questo tipo si verificano solitamente perché è stata omessa la condizione di combinazione. Per le soluzioni consigliate, consultare Loop nidificato.
   - Hash and Hash Join (Hash e Hash Join): gli hash join vengono utilizzati nel caso in cui le tabelle combinate in cui si trovano le colonne di combinazione non siano chiavi di distribuzione né chiavi di ordinamento. Per le soluzioni consigliate, consultare Hash join.
   - Merge Join: non è necessaria alcuna modifica.
3. Nota quale tabella viene utilizzata per le inner join e quale per le outer join. Generalmente, il motore di query sceglie la tabella più piccola per le inner join, la più grande per le outer join. Se questa scelta non avviene, le tue statistiche saranno probabilmente scadute. Per le soluzioni consigliate, consultare Statistiche della tabella mancanti o scadute.

4. Verifica l'esistenza di operazioni di ordinamento ad alto costo. Nel caso in cui ci siano, consultare Righe non ordinate o ordinate in modo errato per le soluzioni consigliate.
5. Cerca i seguenti operatori di trasmissione dove esistono operazioni ad alto costo:
   - DS_BCAST_INNER: Indica che la tabella viene trasmessa a tutti i nodi di calcolo. Questo va bene per una piccola tabella, ma non è ideale per una tabella più grande.
   - DS_DIST_ALL_INNER: indica che tutto il carico di lavoro si trova in una singola sezione.
   - DS_DIST_BOTH: indica una ridistribuzione impegnativa.

   Per le soluzioni consigliate in queste situazioni, consultare Distribuzione dei dati non ottimale.

Analisi del riepilogo della query

Per ottenere fasi di esecuzione e statistiche più dettagliate rispetto a quelle fornite dal piano di query prodotto da EXPLAIN, utilizzare le visualizzazioni di sistema SVL_QUERY_SUMMARY e SVL_QUERY_REPORT.

SVL_QUERY_SUMMARY fornisce le statistiche di query dal flusso. Puoi utilizzare le informazioni fornite per rilevare il problema delle fasi costose, con un'esecuzione lunga e di quelle che scrivono al disco.

La vista di sistema SVL_QUERY_REPORT ti permette di visualizzare le informazioni simili a quelle di SVL_QUERY_SUMMARY, solo tramite le sezioni del nodo di calcolo piuttosto che tramite il flusso. Puoi utilizzare le informazioni a livello di sezione per l'individuazione della distribuzione di dati non uniforme all'interno del cluster (nota anche come differenza di distribuzione dei dati), la quale forza alcuni nodi a lavorare maggiormente rispetto ad altri e compromette le prestazioni della query.

Argomenti

- Utilizzo della visualizzazione SVL_QUERY_SUMMARY
- Utilizzo della vista SVL_QUERY_REPORT
- Mappatura del piano di query sul riepilogo della query

Utilizzo della visualizzazione SVL_QUERY_SUMMARY

Per analizzare le informazioni di riepilogo della query attraverso il flusso, esegui quanto segue:

1. Esegui la seguente query per determinare l'ID della tua query:

```
select query, elapsed, substring
from svl_qlog
order by query
desc limit 5;
```

Esamina il testo troncato della query nel campo `substring` per determinare quale valore `query` rappresenta la tua query. Se hai eseguito la query più di una volta, utilizza il valore `query` a partire dalla riga con il valore `elapsed` più basso. Questa è la riga per la versione compilata. Se stai eseguendo molte query, puoi alzare il valore utilizzato dalla clausola LIMIT, che viene usata per accertarsi che la query sia inclusa.

2. Seleziona le righe da SVL_QUERY_SUMMARY per la tua query. Ordina i risultati per flusso, segmento e fase:

```
select * from svl_query_summary where query = MyQueryID order by stm, seg, step;
```

Result 1 | Messages

userid	query	stm	seg	step	maxtime	avgtime	rows	bytes	rate_row	rate_byte	label	is_diskbased	workmem	is_rrscan	is_delayed_scan	rows_pre_filter
1	249059	0	0	0	58	27	4	192			scan tbl=246 name=Internal Worktable	f	0	f	f	0
1	249059	0	0	1	58	27	4	0			project	f	0	f	f	0
1	249059	0	0	2	58	27	4	64			save tbl=249	f	481296384	f	f	0
1	249059	1	1	0	20	20	1	48			scan tbl=250 name=Internal Worktable	f	0	f	f	0
1	249059	1	1	1	20	20	1	0			dist	f	0	f	f	0
1	249059	1	2	0	2275	1350	1	48			scan tbl=19221 name=Internal Worktable	f	0	f	f	0
1	249059	1	2	1	2275	1350	1	0			project	f	0	f	f	0
1	249059	1	2	2	2275	1350	1	16			save tbl=249	f	475004928	f	f	0
1	249059	2	3	0	1640	792	5	80			scan tbl=249 name=Internal Worktable	f	0	f	f	0
1	249059	2	3	1	1640	792	5	80			sort tbl=248	f	468713472	f	f	0
1	249059	3	4	0	26	9	5	80			scan tbl=248 name=Internal Worktable	f	0	f	f	0
1	249059	3	4	1	26	9	5	0			return	f	0	f	f	0
1	249059	3	5	0	49	49	0	0			merge	f	0	f	f	0
1	249059	3	5	1	49	49	5	0			project	f	0	f	f	0
1	249059	3	5	2	49	49	0	0			return	f	0	f	f	0

3. Per mappare le fasi per le operazioni nel piano di query, consultare le informazioni contenute in Mappatura del piano di query sul riepilogo della query. Dovrebbero avere approssimativamente lo stesso valore per righe e byte (righe * larghezza del piano di query). Altrimenti, consultare Statistiche della tabella mancanti o scadute per le soluzioni consigliate.
4. Verifica che il campo `is_diskbased` abbia un valore `t` (true) per ogni fase. Hash, Aggregate e Sort sono gli operatori che, probabilmente, scriveranno dati al disco nel caso in cui il sistema non abbia abbastanza memoria allocata per l'elaborazione della query.

   Se `is_diskbased` ha il valore "true", consultare Memoria insufficiente allocata alla query per le soluzioni consigliate.
5. Rivedi il valore del campo `label` e verifica se da qualche parte nelle fasi ci sia una sequenza AGG-DIST-AGG. La sua presenza indica l'aggregazione a due fasi, che è molto costosa. Per sistemarla, modifica la clausola GROUP BY per utilizzare la chiave di distribuzione (la prima chiave, nel caso in cui ce ne fossero più di una).

6. Rivedi il valore `maxtime` per ogni segmento (è lo stesso all'interno di tutte le fasi nel segmento). Identifica il segmento con il valore `maxtime` più alto e rivedi le fasi in questo segmento per i seguenti operatori.

   > **Note**
   >
   > Un valore `maxtime` alto non indica necessariamente che ci sia un problema con il segmento. Nonostante abbia un valore alto, il segmento potrebbe non aver impiegato molto tempo per essere elaborato. Tutti i segmenti in un flusso cominciano a programmarsi all'unisono. Tuttavia, è possibile che alcuni segmenti downstream non possano essere eseguiti finché non si ottengono dati da quelli upstream. Questo effetto fa sì che appaia abbiano impiegato più tempo, dato che il loro valore `maxtime` includerà sia il tempo di attesa che quello di elaborazione.

   - BCAST or DIST (BCAST o DIST): in questi casi, il valore `maxtime` elevato può derivare dalla redistribuzione di un gran numero di righe. Per le soluzioni consigliate, consultare Distribuzione dei dati non ottimale.
   - HJOIN (hash join): se la fase in questione ha un valore molto elevato nel campo `rows` rispetto al valore `rows` nella fase finale RETURN nella query, consultare Hash join per le soluzioni consigliate.
   - SCAN/SORT: subito prima di una fase di combinazione, cerca una sequenza di fasi SCAN, SORT, SCAN, MERGE. Questo modello indica che i dati non ordinati sono stati scansionati e uniti all'area ordinata della tabella.

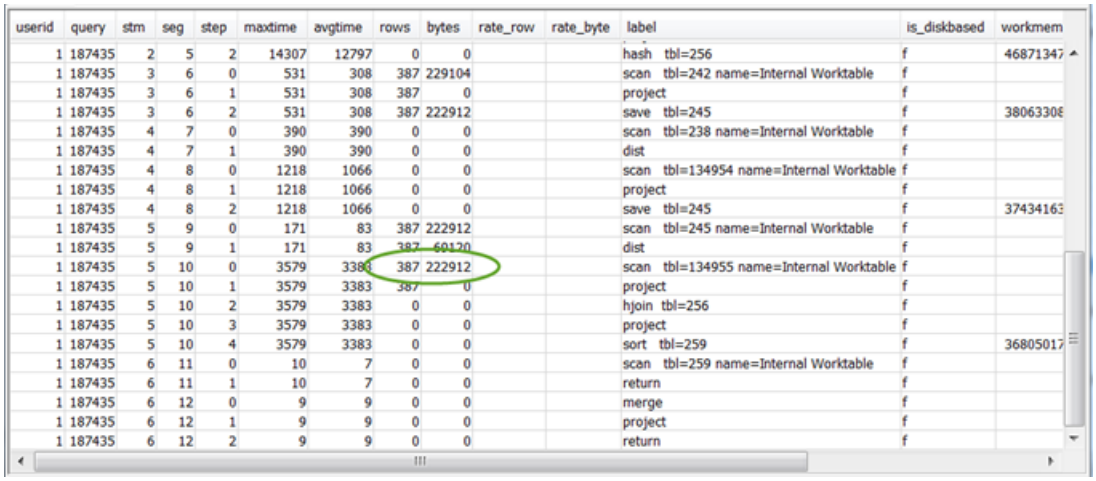
     Verifica che il valore delle righe per la fase SCAN abbia un valore molto più alto rispetto a quello nella fase finale RETURN nella query. Questo modello indica che il motore di esecuzione sta scansionando le file che verranno eliminate, il che è inefficiente. Per le soluzioni consigliate, consultare Predicato poco restrittivo.

     Se il valore `maxtime` della fase SCAN è alto, consultare Clausola WHERE non ottimale per le soluzioni consigliate.

     Se il valore `rows` della fase SORT non è zero, consultare Righe non ordinate o ordinate in modo errato per le soluzioni consigliate.

7. Per avere un'idea del totale dei dati restituiti al client, rivedi i valori `rows` e `bytes` delle fasi 5–10 che precedono la fase finale RETURN. Questo processo può essere un po' un'arte.

Ad esempio, nel seguente riepilogo di query, puoi constatare che la terza fase PROJECT fornisce un valore `rows` ma non un valore `bytes`. Consultando le fasi precedenti per trovarne uno con lo stesso valore `rows`, troverai che la fase SCAN fornisce sia informazioni relative alle righe che ai byte:



userid	query	stm	seg	step	maxtime	avgtime	rows	bytes	rate_row	rate_byte	label	is_diskbased	workmem
1	187435	2	5	2	14307	12797	0	0			hash tbl=256	f	46871347
1	187435	3	6	0	531	308	387	229104			scan tbl=242 name=Internal Worktable	f	
1	187435	3	6	1	531	308	387	0			project	f	
1	187435	3	6	2	531	308	387	222912			save tbl=245	f	3806330
1	187435	4	7	0	390	390	0	0			scan tbl=238 name=Internal Worktable	f	
1	187435	4	7	1	390	390	0	0			dist	f	
1	187435	4	8	0	1218	1066	0	0			scan tbl=134954 name=Internal Worktable	f	
1	187435	4	8	1	1218	1066	0	0			project	f	
1	187435	4	8	2	1218	1066	0	0			save tbl=245	f	37434163
1	187435	5	9	0	171	83	387	222912			scan tbl=245 name=Internal Worktable	f	
1	187435	5	9	1	171	83	387	60120			dist	f	
1	187435	5	10	0	3579	3383	387	222912			scan tbl=134955 name=Internal Worktable	f	
1	187435	5	10	1	3579	3383	387	0			project	f	
1	187435	5	10	2	3579	3383	0	0			hjoin tbl=256	f	
1	187435	5	10	3	3579	3383	0	0			project	f	
1	187435	5	10	4	3579	3383	0	0			sort tbl=259	f	36805017
1	187435	6	11	0	10	7	0	0			scan tbl=259 name=Internal Worktable	f	
1	187435	6	11	1	10	7	0	0			return	f	
1	187435	6	12	0	9	9	0	0			merge	f	
1	187435	6	12	1	9	9	0	0			project	f	
1	187435	6	12	2	9	9	0	0			return	f	

Se stai restituendo un grande volume insolito dei dati, consultare Insieme di risultati molto grande per le soluzioni consigliate.

8. Verifica se per ogni fase il valore `bytes` è alto rispetto al valore `rows` rispetto alle altre fasi. Questo modello può indicare che stai selezionando molte colonne. Per le soluzioni consigliate, consultare Elenco SELECT grande.

Utilizzo della vista SVL_QUERY_REPORT

Per analizzare le informazioni di riepilogo della query attraverso la sezione, esegui quanto segue:

1. Esegui la seguente operazione per determinare l'ID della tua query:

```
select query, elapsed, substring
from svl_qlog
order by query
desc limit 5;
```

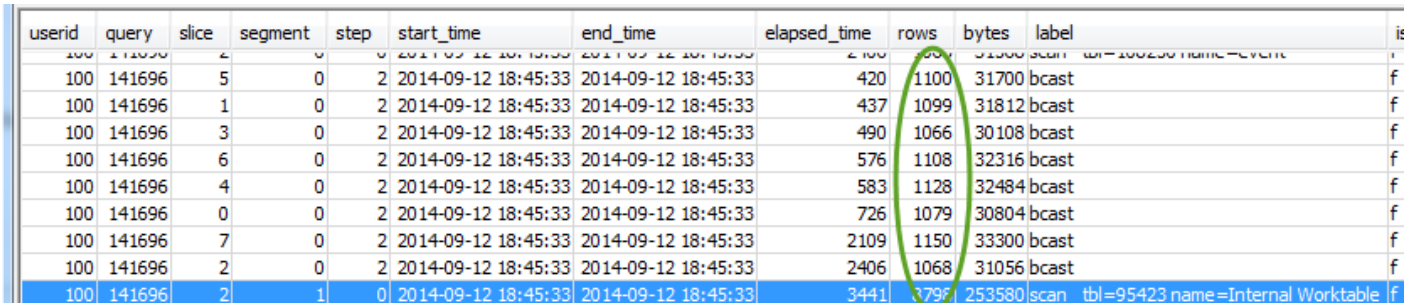
Esamina il testo troncato della query nel campo `substring` per determinare quale valore `query` rappresenta la tua query. Se hai eseguito la query più di una volta, utilizza il valore `query` a partire dalla riga con il valore `elapsed` più basso. Questa è la riga per la versione compilata. Se stai

eseguendo molte query, puoi alzare il valore utilizzato dalla clausola LIMIT, che viene usata per accertarsi che la query sia inclusa.

2. Seleziona le righe da SVL_QUERY_REPORT per la tua query. Ordina i risultati per segmento, fase, tempo trascorso e righe:

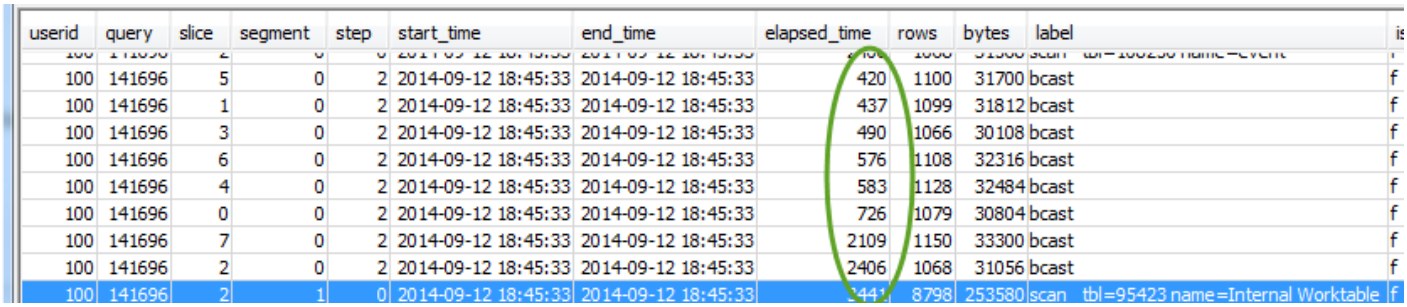
```
select * from svl_query_report where query = MyQueryID order by segment, step,
 elapsed_time, rows;
```

3. Per ogni fase, verifica che tutte le sezioni abbiano processato approssimativamente lo stesso numero di righe:



userid	query	slice	segment	step	start_time	end_time	elapsed_time	rows	bytes	label	is
100	141696	5	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	420	1100	31700	bcast	f
100	141696	1	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	437	1099	31812	bcast	f
100	141696	3	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	490	1066	30108	bcast	f
100	141696	6	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	576	1108	32316	bcast	f
100	141696	4	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	583	1128	32484	bcast	f
100	141696	0	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	726	1079	30804	bcast	f
100	141696	7	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	2109	1150	33300	bcast	f
100	141696	2	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	2406	1068	31056	bcast	f
100	141696	2	1	0	2014-09-12 18:45:33	2014-09-12 18:45:33	3441	8798	253580	scan tbl=95423 name=Internal Worktable	f

Verifica anche che tutte le sezioni abbiano impiegato approssimativamente lo stesso tempo:



userid	query	slice	segment	step	start_time	end_time	elapsed_time	rows	bytes	label	is
100	141696	5	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	420	1100	31700	bcast	f
100	141696	1	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	437	1099	31812	bcast	f
100	141696	3	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	490	1066	30108	bcast	f
100	141696	6	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	576	1108	32316	bcast	f
100	141696	4	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	583	1128	32484	bcast	f
100	141696	0	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	726	1079	30804	bcast	f
100	141696	7	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	2109	1150	33300	bcast	f
100	141696	2	0	2	2014-09-12 18:45:33	2014-09-12 18:45:33	2406	1068	31056	bcast	f
100	141696	2	1	0	2014-09-12 18:45:33	2014-09-12 18:45:33	3441	8798	253580	scan tbl=95423 name=Internal Worktable	f

Grosse discrepanze in questi valori possono indicare che, la differenza della distribuzione dei dati, è dovuta allo stile di distribuzione non ottimale per questa query particolare. Per le soluzioni consigliate, consultare Distribuzione dei dati non ottimale.

Mappatura del piano di query sul riepilogo della query

Aiuta a mappare le operazioni dal piano di query alle fasi (identificate dai valori del campo dell'etichetta) nel riepilogo della query, al fine di ottenere maggiori dettagli su di esse:

Operazione del piano di query	Valore del campo etichetta	Descrizione
Aggregazione HashAggregate GroupAggregate	AGGR	Valuta le funzioni di aggregazione e le condizioni GROUP BY.
DS_BCAST_INNER	BCAST (trasmissione)	Trasmette un'intera tabella o alcun insieme di righe (come un insieme filtrato di righe da una tabella) a tutti i nodi.
Non viene mostrato nel piano di query	DELETE	Elimina le righe dalle tabelle.
DS_DIST_NONE DS_DIST_ALL_NONE DS_DIST_INNER DS_DIST_ALL_INNER DS_DIST_ALL_BOTH	DIST (distribuzione)	Distribuisce le righe ai nodi per combinazioni parallele o per altre elaborazioni parallele.
HASH	HASH	Costruisce tabelle hash da utilizzare sugli hash join.
Hash join	HJOIN (hash join)	Esegue un hash join delle due tabelle o degli insiemi di risultati intermedi.
Non viene mostrato nel piano di query	INSERT	Inserisce le righe nelle tabelle.
Limite	LIMIT	Applica una clausola LIMIT agli insiemi dei risultati.

Operazione del piano di query	Valore del campo etichetta	Descrizione
Unione	MERGE	Unisce le righe prodotte dalle operazioni di combinazione o di ordinamento parallelo.
Merge join	MJOIN (merge join)	Esegue un merge join delle due tabelle o degli insiemi di risultati intermedi.
Loop nidificato	NLOOP (loop nidificato)	Esegue un loop nidificato delle due tabelle o degli insiemi di risultati intermedi.
Non viene mostrato nel piano di query	PARSE	Analizza le stringhe in valori binari per il caricamento.
Progetto	PROJECT	Valuta le espressioni
Rete	RETURN	Restituisce le righe al nodo principale o al client.
Non viene mostrato nel piano di query	SAVE	Materializza le righe da utilizzare nella prossima fase di elaborazione.
Seq Scan	SCAN	Scansiona le tabelle o gli insiemi di risultati intermedi.
Ordina	SORT	Ordina le righe o gli insiemi di risultati intermedi come richiesto da altre operazion i successive (ad esempio aggregazioni e combinazi oni), oppure le ordina al fine di soddisfare una clausola ORDER BY.

Operazione del piano di query	Valore del campo etichetta	Descrizione
Unique	UNIQUE	Applica una clausola SELECT DISTINCT o rimuove i duplicati, come richiesto da altre operazioni.
Window	WINDOW	Calcola le funzioni di aggregazione e di classific azione della finestra.

Miglioramento delle prestazioni della query di

Di seguito vengono riportati problemi comuni che riguardano le prestazioni delle query, con istruzioni su come diagnosticarli e risolverli.

Argomenti

Statistiche della tabella mancanti o scadute

Se le statistiche della tabella sono mancanti o scadute, potresti visualizzare quanto segue:

- Un messaggio di avviso nei risultati del comando EXPLAIN.

- Un evento di avviso statistiche mancanti su STL_ALERT_EVENT_LOG. Per ulteriori informazioni, consultare Revisione degli avvisi di query.

Per risolvere questo problema, esegui il comando ANALYZE.

Loop nidificato

Se è presente un loop nidificato, potresti visualizzare un evento di avviso loop nidificato su STL_ALERT_EVENT_LOG. Per identificare questo tipo di evento, puoi anche eseguire la query su Identificazione di query con loop nidificati. Per ulteriori informazioni, consultare Revisione degli avvisi di query.

Per risolvere questo problema, rivedi la tua query per le combinazioni incrociate e rimuovile se possibile. Le combinazioni incrociate sono combinazione senza una condizione di combinazione nel prodotto cartesiano delle due tabelle. Generalmente vengono eseguite come combinazioni di loop nidificati, che sono le più lente tra i tipi di combinazione possibili.

Hash join

Se è presente un hash join, potresti visualizzare quanto segue:

- Le operazioni hash e hash join nel piano di query. Per ulteriori informazioni, consultare Analisi del piano di query.
- Una fase HJOIN nel segmento con il valore maxtime più alto su SVL_QUERY_SUMMARY. Per ulteriori informazioni, consultare Utilizzo della visualizzazione SVL_QUERY_SUMMARY.

Il problema può essere risolto in diversi modi:

- Se possibile, riscrivi la query per utilizzare un merge join. Puoi farlo mediante la specificazione di quelle colonne di combinazione che sono sia chiavi di distribuzione che chiavi di ordinamento.
- Se la fase HJOIN su SVL_QUERY_SUMMARY ha un valore molto alto nel campo relativo alle righe rispetto al valore delle righe nella fase finale RETURN nella query, verifica se puoi riscrivere la query per combinarla su un'unica colonna. Se una query non viene combinata su un'unica colonna, ad esempio una chiave primaria, aumenta il numero di righe coinvolte nella combinazione.

Righe fantasma o righe di cui non è stato eseguito il commit

Se sono presenti righe fantasma o non eseguite, potresti visualizzare un evento di avviso su STL_ALERT_EVENT_LOG che indica l'eccessiva presenza di righe fantasma. Per ulteriori informazioni, consultare Revisione degli avvisi di query.

Il problema può essere risolto in diversi modi:

- Cercare nella scheda Carichi della console Amazon Redshift le operazioni di carico attive su tutte le tabelle della query. Se visualizzi operazioni di carico attive, attendi il loro completamento prima di intraprendere altre operazioni.
- Se non ci sono operazioni di carico attive, esegui il comando VACUUM sulle tabelle della query al fine di rimuovere le righe eliminate.

Righe non ordinate o ordinate in modo errato

Se sono presenti delle righe non ordinate o ordinate male, potresti visualizzare un evento di avviso del filtro molto selettivo su STL_ALERT_EVENT_LOG. Per ulteriori informazioni, consultare Revisione degli avvisi di query.

Inoltre, puoi verificare la presenza di grandi aree non ordinate nelle tabelle della tua query, eseguendo la stessa su Identificazione delle tabelle con differenza di dati o con righe non ordinate.

Il problema può essere risolto in diversi modi:

- Esegui il comando VACUUM sulle tabelle della query per riordinare le righe.
- Per verificare se è possibile apportare dei miglioramenti, rivedi la chiave di ordinamento sulle tabelle della query. Ricordati di comparare le prestazioni di questa query a quelle di altre query importanti e al sistema globale prima di apportare delle modifiche. Per ulteriori informazioni, consultare Utilizzo delle chiavi di ordinamento.

Distribuzione dei dati non ottimale

Se la distribuzione dei dati non è ottimale, potresti visualizzare quanto segue:

- Su STL_ALERT_EVENT_LOG apparirà un evento di avviso di una grande trasmissione, distribuzione o un'esecuzione seriale. Per ulteriori informazioni, consultare Revisione degli avvisi di query.

- Le sezioni non stanno processando approssimativamente lo stesso numero di righe per una fase specifica. Per ulteriori informazioni, consultare Utilizzo della vista SVL_QUERY_REPORT.
- Le sezioni non stanno impiegando approssimativamente lo stesso tempo per una fase specifica. Per ulteriori informazioni, consultare Utilizzo della vista SVL_QUERY_REPORT.

Se nessuna delle opzioni precedenti è vera, puoi anche verificare se qualche tabella nella tua query presenta una differenza di dati, eseguendo la stessa su Identificazione delle tabelle con differenza di dati o con righe non ordinate.

Per risolvere questo problema, esamina gli stili di distribuzione delle tabelle nella query e determina se è possibile apportare dei miglioramenti. Ricordati di comparare le prestazioni di questa query a quelle di altre query importanti e al sistema globale prima di apportare delle modifiche. Per ulteriori informazioni, consultare Utilizzo degli stili di distribuzione dati.

Memoria insufficiente allocata alla query

Se la memoria allocata alla tua query è insufficiente, potresti visualizzare una fase su SVL_QUERY_SUMMARY che dispone di un valore "true" su `is_diskbased`. Per ulteriori informazioni, consultare Utilizzo della visualizzazione SVL_QUERY_SUMMARY.

Per risolvere il problema, alloca più memoria alla query aumentando temporaneamente il numero di slot che utilizza la query. La gestione del carico di lavoro (WLM) riserva degli slot in una coda di query equivalente per il livello di simultaneità impostato per la query. Ad esempio, una coda con un livello di simultaneità pari a 5, dispone di 5 slot. La memoria assegnata alla coda viene allocata in parti uguali a ogni slot. L'assegnazione di più slot per una query, le fornisce l'accesso alla memoria di tutti questi slot. Per ulteriori informazioni su come incrementare temporaneamente gli slot di una query, consultare wlm_query_slot_count.

Clausola WHERE non ottimale

Se la tua clausola WHERE causa un'eccessiva scansione della tabella, potresti visualizzare una fase SCAN nel segmento con il valore `maxtime` più alto su SVL_QUERY_SUMMARY. Per ulteriori informazioni, consultare Utilizzo della visualizzazione SVL_QUERY_SUMMARY.

Per risolvere questo problema, aggiungi una clausola WHERE alla query basata sulla colonna di ordinamento principale della tabella più grande. Grazie a questo approccio, è possibile ridurre al minimo i tempi di scansione. Per ulteriori informazioni, consultare Best practice di Amazon Redshift per la progettazione di tabelle.

Predicato poco restrittivo

Se la tua query dispone di un predicato poco restrittivo, potresti visualizzare una fase SCAN nel segmento con il valore `maxtime` più alto su SVL_QUERY_SUMMARY, il quale ha un valore `rows` molto alto rispetto al valore `rows` nella fase finale RETURN nella query. Per ulteriori informazioni, consultare Utilizzo della visualizzazione SVL_QUERY_SUMMARY.

Per risolvere questo problema. prova ad aggiungere un predicato alla query, oppure rendi il predicato esistente più restrittivo per restringere l'output.

Insieme di risultati molto grande

Se la query restituisce un set di risultati molto ampio, considerare la riscrittura della query per utilizzare UNLOAD e scrivere i risultati in Amazon S3. Questo approccio migliorerà le prestazioni della fase RETURN, traendo vantaggio dall'elaborazione parallela. Per ulteriori informazioni sulla verifica di un insieme di risultati molto grande, consultare Utilizzo della visualizzazione SVL_QUERY_SUMMARY.

Elenco SELECT grande

Se la tua query dispone di un elenco SELECT insolitamente grande, potresti visualizzare un valore `bytes` più alto rispetto al valore `rows` di tutte le fasi (in relazione alle altre fasi) su SVL_QUERY_SUMMARY. Questo valore `bytes` alto può indicare che stai selezionando molte colonne. Per ulteriori informazioni, consultare Utilizzo della visualizzazione SVL_QUERY_SUMMARY.

Per risolvere questo problema, rivedi le colonne che stai selezionando e verifica se puoi rimuoverne qualcuna.

Query di diagnostica per l'ottimizzazione di query

Utilizza le seguenti query per rilevare problemi con le query o con le loro tabelle sottostanti, che possono influenzare le prestazioni delle query. Consigliamo l'uso di queste query in combinazione con i processi di ottimizzazione della query illustrati su Analisi e miglioramento delle query.

Argomenti

- Identificazione delle migliori query per l'ottimizzazione
- Identificazione delle tabelle con differenza di dati o con righe non ordinate
- Identificazione di query con loop nidificati

- Revisione dei tempi di attesa della coda per le query
- Revisione degli avvisi di query per tabella
- Identificazione delle tabelle con statistiche mancanti

Identificazione delle migliori query per l'ottimizzazione

Le seguenti query identificano le 50 istruzioni più dispendiose in termini di tempo, che sono state eseguite negli ultimi 7 giorni. È possibile utilizzare i risultati per identificare le query che impiegano un tempo insolitamente lungo. È possibile anche identificare le query che vengono eseguite frequentemente (ovvero quelle che appaiono più di una volta nell'insieme di risultati). Queste sono spesso buone query da ottimizzare per migliorare le prestazioni del sistema.

Questa query fornisce inoltre un numero di eventi di avviso associati a ogni query rilevata. Questi avvisi forniscono dettagli che puoi utilizzare per migliorare le prestazioni della query. Per ulteriori informazioni, consultare Revisione degli avvisi di query.

```
select trim(database) as db, count(query) as n_qry,
max(substring (qrytext,1,80)) as qrytext,
min(run_minutes) as "min" ,
max(run_minutes) as "max",
avg(run_minutes) as "avg", sum(run_minutes) as total,
max(query) as max_query_id,
max(starttime)::date as last_run,
sum(alerts) as alerts, aborted
from (select userid, label, stl_query.query,
trim(database) as database,
trim(querytxt) as qrytext,
md5(trim(querytxt)) as qry_md5,
starttime, endtime,
(datediff(seconds, starttime,endtime)::numeric(12,2))/60 as run_minutes,
alrt.num_events as alerts, aborted
from stl_query
left outer join
(select query, 1 as num_events from stl_alert_event_log group by query ) as alrt
on alrt.query = stl_query.query
where userid <> 1 and starttime >=  dateadd(day, -7, current_date))
group by database, label, qry_md5, aborted
order by total desc limit 50;
```

Identificazione delle tabelle con differenza di dati o con righe non ordinate

La seguente query identifica le tabelle che dispongono di una distribuzione di dati non uniforme (differenza di dati) o di un'alta percentuale di righe non ordinate.

Un valore `skew` basso indica che i dati della tabella sono distribuiti in modo idoneo. Se una tabella ha un valore `skew` maggiore o uguale a 4.00, considera la modifica del suo stile di distribuzione dei dati. Per ulteriori informazioni, consultare Distribuzione dei dati non ottimale.

Se una tabella ha un valore `pct_unsorted` maggiore del 20 percento, considera l'esecuzione del comando VACUUM. Per ulteriori informazioni, consultare Righe non ordinate o ordinate in modo errato.

Dovresti anche rivedere i valori `mbytes` e `pct_of_total` di ogni tabella. Queste colonne identificano le dimensioni della tabella e la percentuale dello spazio del disco non elaborato che consuma la tabella. Questo spazio include lo spazio riservato da Amazon Redshift per uso interno, di conseguenza è superiore alla capacità nominale del disco, ovvero lo spazio su disco disponibile per l'utente. Utilizza queste informazioni per essere sicuro di disporre di uno spazio libero del disco pari ad almeno due volte e mezza la dimensione della tabella più grande. La disponibilità di questo spazio, permette al sistema di scrivere risultati intermedi sul disco quando si elaborano query complesse.

```
select trim(pgn.nspname) as schema,
trim(a.name) as table, id as tableid,
decode(pgc.reldiststyle,0, 'even',1,det.distkey ,8,'all') as distkey,
 dist_ratio.ratio::decimal(10,4) as skew,
det.head_sort as "sortkey",
det.n_sortkeys as "#sks", b.mbytes,
decode(b.mbytes,0,0,((b.mbytes/part.total::decimal)*100)::decimal(5,2)) as
 pct_of_total,
decode(det.max_enc,0,'n','y') as enc, a.rows,
decode( det.n_sortkeys, 0, null, a.unsorted_rows ) as unsorted_rows ,
decode( det.n_sortkeys, 0, null, decode( a.rows,0,0, (a.unsorted_rows::decimal(32)/
a.rows)*100) )::decimal(5,2) as pct_unsorted
from (select db_id, id, name, sum(rows) as rows,
sum(rows)-sum(sorted_rows) as unsorted_rows
from stv_tbl_perm a
group by db_id, id, name) as a
join pg_class as pgc on pgc.oid = a.id
join pg_namespace as pgn on pgn.oid = pgc.relnamespace
left outer join (select tbl, count(*) as mbytes
```

```
from stv_blocklist group by tbl) b on a.id=b.tbl
inner join (select attrelid,
min(case attisdistkey when 't' then attname else null end) as "distkey",
min(case attsortkeyord when 1 then attname  else null end ) as head_sort ,
max(attsortkeyord) as n_sortkeys,
max(attencodingtype) as max_enc
from pg_attribute group by 1) as det
on det.attrelid = a.id
inner join ( select tbl, max(mbytes)::decimal(32)/min(mbytes) as ratio
from (select tbl, trim(name) as name, slice, count(*) as mbytes
from svv_diskusage group by tbl, name, slice )
group by tbl, name ) as dist_ratio on a.id = dist_ratio.tbl
join ( select sum(capacity) as  total
from stv_partitions where part_begin=0 ) as part on 1=1
where mbytes is not null
order by  mbytes desc;
```

Identificazione di query con loop nidificati

La query seguente identifica le query che hanno registrato eventi di avviso per loop nidificati. Per informazioni su come risolvere la condizione del loop nidificato, consultare Loop nidificato.

```
select query, trim(querytxt) as SQL, starttime
from stl_query
where query in (
select distinct query
from stl_alert_event_log
where event like 'Nested Loop Join in the query plan%')
order by starttime desc;
```

Revisione dei tempi di attesa della coda per le query

La seguente query illustra il tempo di attesa di query recenti per aprire uno slot nella coda della query, prima che potessero essere eseguite. Se i tempi di attesa sono tendenzialmente alti, potresti voler modificare la configurazione della coda della query per un migliore throughput. Per ulteriori informazioni, consultare Implementazione di WLM manuale.

```
select trim(database) as DB , w.query,
substring(q.querytxt, 1, 100) as querytxt,  w.queue_start_time,
w.service_class as class, w.slot_count as slots,
w.total_queue_time/1000000 as queue_seconds,
```

```
w.total_exec_time/1000000 exec_seconds, (w.total_queue_time+w.total_Exec_time)/1000000
 as total_seconds
from stl_wlm_query w
left join stl_query q on q.query = w.query and q.userid = w.userid
where w.queue_start_Time >= dateadd(day, -7, current_Date)
and w.total_queue_Time > 0  and w.userid >1
and q.starttime >= dateadd(day, -7, current_Date)
order by w.total_queue_time desc, w.queue_start_time desc limit 35;
```

Revisione degli avvisi di query per tabella

La query seguente identifica le tabella che hanno registrato eventi di avviso e, inoltre, identifica che tipo di avvisi vengono attivati più di frequente.

Se il valore `minutes` di una riga con una tabella identificata è alto, verifica se è necessario eseguire manutenzione di routine, su questa tabella, come l'esecuzione del comando ANALYZE o VACUUM.

Se il valore `count` di una riga è alto ma il valore `table` è nullo, eseguire una query con STL_ALERT_EVENT_LOG per il valore `event` associato, al fine di investigare sul motivo per il quale l'avviso viene attivato così di frequente.

```
select trim(s.perm_table_name) as table,
(sum(abs(datediff(seconds, s.starttime, s.endtime)))/60)::numeric(24,0) as minutes,
 trim(split_part(l.event,':',1)) as event,  trim(l.solution) as solution,
max(l.query) as sample_query, count(*)
from stl_alert_event_log as l
left join stl_scan as s on s.query = l.query and s.slice = l.slice
and s.segment = l.segment and s.step = l.step
where l.event_time >=  dateadd(day, -7, current_Date)
group by 1,3,4
order by 2 desc,6 desc;
```

Identificazione delle tabelle con statistiche mancanti

La seguente query fornisce un conteggio delle query che stai eseguendo con tabelle a cui mancano delle statistiche. Se questa query restituisce delle righe, controlla il valore `plannode` per determinare le tabelle in questione, quindi esegui su di esse il comando ANALYZE.

```
select substring(trim(plannode),1,100) as plannode, count(*)
from stl_explain
where plannode like '%missing statistics%'
```

```
group by plannode
order by 2 desc;
```

Risoluzione dei problemi delle query

Questa sezione fornisce un riferimento rapido per l'identificazione e l'indirizzamento dei problemi più comuni e più gravi che, probabilmente, riscontrerai con le query di Amazon Redshift.

Argomenti

Questi suggerimenti ti forniscono un punto di inizio per la risoluzione dei problemi. Per ulteriori informazioni inoltre, puoi fare riferimento alle seguenti risorse.

- Accesso ai cluster e ai database di Amazon Redshift
- Utilizzo dell'ottimizzazione automatica delle tabelle
- Caricamento dei dati
- Tutorial: Caricamento dei dati da Amazon S3

Connessione non riuscita

La connessione della tua query può avere esito negativo per le seguenti ragioni; suggeriamo i seguenti approcci per la risoluzione di problemi.

Il client non riesce a connettersi al server

Se stai utilizzando certificati di server o SSL, prima rimuovi questa complessità mentre risolvi il problema relativo alla connessione. Quando trovi una soluzione, aggiungi nuovamente i certificati di server o SSL. Per ulteriori informazioni, consulta Configurazione delle opzioni di sicurezza per le connessioni nella Guida alla gestione di Amazon Redshift.

La connessione è rifiutata

Generalmente, quando ricevi un messaggio di errore che indica esito negativo per stabilire una connessione, ciò significa che c'è un problema con l'autorizzazione per l'accesso al cluster. Per ulteriori informazioni, consulta Connessione rifiutata o non riuscita nella Guida alla gestione di Amazon Redshift.

Interruzioni della query

La tua query può interrompersi o bloccarsi per le seguenti ragioni; suggeriamo i seguenti approcci per la risoluzione di problemi.

Connessione al database interrotta

Riduci la dimensione dell'unità di trasmissione massima (MTU). La dimensione dell'MTU determina la dimensione massima, espressa in byte, di un pacchetto che può essere trasferito in un quadro Ethernet dalla tua connessione di rete. Per ulteriori informazioni, consulta Connessione al database interrotta nella Guida alla gestione di Amazon Redshift.

Connessione al database scaduta

La connessione del tuo client al database sembra scadere o interrompersi durante l'esecuzione di query lunghe, come il comando COPY. In questo caso, è possibile che la console Amazon Redshift mostri che la query è stata completata, ma lo strumento del client stesso sembra che la stia ancora eseguendo. I risultati della query potrebbero essere mancanti o incompleti, a seconda del momento in cui la connessione si è interrotta. Questo effetto si presenta quando connessioni inattive vengono terminate da un componente di rete intermedio. Per ulteriori informazioni, consulta Problema di timeout del firewall nella Guida alla gestione di Amazon Redshift.

Si verifica un out-of-memory errore sul lato client con ODBC

Se l'applicazione client utilizza una connessione ODBC e la query crea un set di risultati che è troppo grande per rientrare nella memoria, puoi eseguire il flusso del set di risultati all'applicazione client utilizzando un cursore. Per ulteriori informazioni, consultare DECLARE e Considerazioni sulle prestazioni quando si utilizzano i cursori.

Si verifica un errore sul lato client con JDBC out-of-memory

Quando si tenta di recuperare set di risultati di grandi dimensioni tramite una connessione JDBC, è possibile che si verifichino errori sul lato client. out-of-memory Per ulteriori informazioni, consulta Impostazione del parametro delle dimensioni del recupero JDBC.

Blocco potenziale

Se esiste un blocco potenziale, prova quanto segue:

- Visualizza le tabelle di sistema STV_LOCKS e STL_TR_CONFLICT per individuare conflitti sugli aggiornamenti di più di una tabella.
- Utilizza la funzione PG_CANCEL_BACKEND per cancellare una o più query di conflitto.
- Utilizza la funzione PG_TERMINATE_BACKEND per terminare una sessione. Questa operazione forza qualsiasi transazione attualmente in esecuzione nella sessione terminata, al fine di rilasciare tutti i blocchi ed eseguire il rollback della transazione.
- Programma attentamente le operazioni di scrittura simultanee. Per ulteriori informazioni, consultare Gestione delle operazioni di scrittura simultanee.

La query impiega troppo tempo

La tua query può impiegare troppo tempo per le seguenti ragioni; suggeriamo i seguenti approcci per la risoluzione di problemi.

Le tabelle non sono ottimizzate

Per ricevere tutti i vantaggi dell'elaborazione parallela, imposta la chiave di ordinamento, lo stile di distribuzione e la codifica della compressione delle tabelle. Per ulteriori informazioni, consultare Utilizzo dell'ottimizzazione automatica delle tabelle

La query sta scrivendo sul disco

Le tue query potrebbero scrivere al disco per almeno una parte dell'esecuzione della query. Per ulteriori informazioni, consultare Miglioramento delle prestazioni della query di .

La query deve attendere il completamento di altre query

Attraverso la creazione di code della query e l'assegnazione di differenti tipi di query per code idonee, potresti essere in grado di migliorare le prestazioni complessive del sistema. Per ulteriori informazioni, consultare Implementazione della gestione del carico di lavoro.

Le query non sono ottimizzate

Analizza il piano di spiegazione per scoprire opportunità di riscrittura delle query o di ottimizzazione del database. Per ulteriori informazioni, consultare Piano di query.

Le query necessitano di più memoria per essere eseguite

Se una query specifica richiede più memoria, puoi aumentare la memoria disponibile aumentando wlm_query_slot_count.

Il database richiede un comando VACUUM per essere eseguito

Esegui il comando VACUUM quando aggiungi, elimini o modifichi un gran numero di righe, a meno che non carichi i tuoi dati nell'ordine della chiave di ordinamento. Il comando VACUUM riorganizza i tuoi dati al fine di mantenere l'ordine e ripristinare le prestazioni. Per ulteriori informazioni, consulta Vacuum delle tabelle.

Risorse aggiuntive per la risoluzione di query di lunga durata

Di seguito sono riportati gli argomenti relativi alla visualizzazione del sistema e altre sezioni della documentazione utili per l'ottimizzazione delle query:

- La vista di sistema STV_INFLIGHT mostra quali query sono in esecuzione sul cluster. Può essere utile utilizzarlo insieme a STV_RECENTS per determinare quali interrogazioni sono attualmente in esecuzione o completate di recente.
- SYS_QUERY_HISTORY è utile per la risoluzione dei problemi. Mostra le query DDL e DML con proprietà pertinenti come il loro stato attuale, ad esempio `running` o `failed`, il tempo impiegato per l'esecuzione di ciascuna di esse e l'eventuale esecuzione di una query su un cluster con scalabilità concorrenziale.
- STL_QUERYTEXT acquisisce il testo di query per i comandi SQL.
  Inoltre, SVV_QUERY_INFLIGHT, che unisce STL_QUERYTEXT a STV_INFLIGHT, mostra più metadati delle query.
- Un conflitto di blocco delle transazioni può essere una possibile origine di problemi di prestazioni delle query. Per informazioni sulle transazioni attualmente bloccate sulle tabelle, consulta SVV_TRANSACTIONS.
- Identificazione delle domande più adatte al tuning fornisce una query di risoluzione dei problemi che consente di determinare quali query eseguite di recente hanno richiesto più tempo. Questo può aiutarti a concentrare i tuoi sforzi sulle domande che richiedono miglioramenti.

- Se desideri approfondire la gestione delle query e capire come gestire le code di query, Implementazione della gestione del carico di lavoro mostra come farlo. La gestione del carico di lavoro è una funzionalità avanzata e nella maggior parte dei casi consigliamo la gestione automatizzata del carico di lavoro.

Caricamento non riuscito

Il caricamento dei tuoi dati può avere esito negativo per le seguenti ragioni; suggeriamo i seguenti approcci per la risoluzione di problemi.

L'origine dei dati si trova in una regione diversa AWS

Per impostazione predefinita, il bucket Amazon S3 o la tabella Amazon DynamoDB specificati nel comando COPY devono trovarsi nella stessa regione del cluster. AWS Se i dati e il cluster si trovano in regioni differenti, riceverai un messaggio di errore simile al seguente:

```
The bucket you are attempting to access must be addressed using the specified endpoint.
```

Se possibile, assicurati che il cluster e l'origine dei dati si trovino nella stessa regione. Puoi specificare una regione diversa utilizzando l'opzione REGION del comando COPY.

> **Note**
>
> Se il cluster e la fonte di dati si trovano in AWS regioni diverse, sono previsti costi di trasferimento dei dati. Avrai anche una latenza maggiore.

Esito negativo del comando COPY

Eseguire una query STL_LOAD_ERRORS per scoprire gli errori che si sono verificati durante carichi specifici. Per ulteriori informazioni, consultare STL_LOAD_ERRORS.

Il caricamento impiega troppo tempo

La tua operazione di caricamento può impiegare troppo tempo per le seguenti ragioni; suggeriamo i seguenti approcci per la risoluzione di problemi.

Caricamento dei dati da un singolo file mediante il comando COPY

Dividi i tuoi dati di caricamento in più file. Quando tutti i dati vengono caricati da un singolo file di grandi dimensioni, Amazon Redshift viene forzato a eseguire un caricamento serializzato, che è molto più lento. Il numero dei file dovrebbe essere un multiplo del numero delle sezioni nel tuo cluster; inoltre i file dovrebbero essere di dimensioni simili, tra 1 MB e 1 GB dopo la compressione. Per ulteriori informazioni, consultare Best practice di Amazon Redshift per la progettazione di query.

Le operazioni di carica utilizzano più comandi COPY

Se vengono utilizzati contemporaneamente più comandi COPY per caricare una tabella da più file, Amazon Redshift viene forzato a eseguire un caricamento serializzato, che è più lento. In questo caso, utilizza un singolo comando COPY.

Dati di caricamento sbagliati

L'operazione COPY può caricare dati sbagliati nei seguenti modi; suggeriamo i seguenti approcci per la risoluzione di problemi.

Vengono caricati file sbagliati

L'utilizzo di un prefisso dell'oggetto per specificare i file dei dati può causare la lettura di file indesiderati. Quindi, utilizza un file manifest per specificare con esattezza i file da caricare. Per ulteriori informazioni, consultare l'opzione copy_from_s3_manifest_file del comando COPY e l'opzione Example: COPY from Amazon S3 using a manifest nell'esempio COPY.

Impostazione del parametro delle dimensioni del recupero JDBC

Per impostazione predefinita, il driver JDBC raccoglie tutti i risultati di una query in una sola volta. Di conseguenza, quando si tenta di recuperare un set di risultati di grandi dimensioni tramite una connessione JDBC, è possibile che si verifichi un errore sul lato client. out-of-memory Per consentire al client di recuperare i set di risultati in batch anziché in un singolo recupero, impostate il parametro all-or-nothing JDBC fetch size nell'applicazione client.

> **Note**
>
> La dimensione del recupero non è supportata da ODBC.

Per le migliori prestazioni, imposta la dimensione del recupero sul valore più alto che non porti a errori di esaurimento della memoria. Un valore della dimensione del recupero più basso causa più

viaggi del server, quindi tempi di esecuzione prolungati. Il server riserva le risorse, tra cui lo slot della query WLM e la memoria associata, fino al momento in cui il client recupera tutto l'insieme di risultati o la query viene cancellata. Quando ottimizzi in modo appropriato la dimensione del recupero, queste risorse vengono rilasciate più velocemente rendendole disponibili alle altre query.

> **Note**
>
> Se devi estrarre set di dati di grandi dimensioni, ti consigliamo di utilizzare un'istruzione UNLOAD per trasferire i dati su Amazon S3. Quando usi UNLOAD, i nodi di calcolo lavorano in parallelo per velocizzare il trasferimento dei dati.

Per ulteriori informazioni sull'impostazione del parametro della dimensione del recupero di JDBC, consultare Ottenimento di risultati basato su un cursore nella documentazione PostgreSQL.

Implementazione della gestione del carico di lavoro

Puoi usare la gestione del carico di lavoro (WLM, workload management) per definire più code di query e instradare le query alle code appropriate in fase di runtime.

In alcuni casi, potrebbero esserci più sessioni o utenti che eseguono query contemporaneamente. In questi casi, alcune query potrebbero consumare risorse cluster per lunghi periodi di tempo e influire sulle prestazioni di altre query. Ad esempio, supponiamo che un gruppo di utenti invii occasionalmente query complesse a esecuzione prolungata che selezionano e ordinano le righe da diverse tabelle di grandi dimensioni. Supponiamo anche che un altro gruppo invii frequentemente query brevi che selezionano solo poche righe da una o due tabelle e che vengono eseguite in pochi secondi. In questa situazione, le query a esecuzione breve potrebbero dover attendere in coda il completamento della query a esecuzione prolungata. WLM aiuta a gestire questa situazione.

È possibile configurare la funzionalità WLM di Amazon Redshift affinché venga eseguita in modalità automatica o manuale.

WLM automatico

Per massimizzare la velocità effettiva del sistema e utilizzare le risorse in modo efficace, è possibile consentire ad Amazon Redshift di gestire il modo in cui le risorse sono divise per l'esecuzione simultanea di query con la WLM automatica. La gestione del carico di lavoro (WLM) automatica gestisce le risorse necessarie per eseguire query. Amazon Redshift determina la quantità di query eseguite simultaneamente e la quantità di memoria allocata a ogni query inviata. È possibile abilitare la WLM automatica utilizzando la console Amazon Redshift scegliendo Cambia modalità WLM, quindi WLM automatica. In questo modo, vengono utilizzate fino a otto code per gestire le query ed entrambi i campi Memory (Memoria) e Concurrency on main (Simultaneità su principale) sono impostati su auto. Puoi specificare una priorità che rispecchi la priorità aziendale del carico di lavoro o degli utenti mappati a ogni coda. La priorità predefinita delle query è impostata su Normal (Normale). Per informazioni su come modificare la priorità delle query in una coda, consultare Priorità delle query. Per ulteriori informazioni, consultare Implementazione del WLM automatico.

In fase di runtime, puoi indirizzare le query a queste code in base ai gruppi di utenti o ai gruppi di query. È inoltre possibile configurare una regola di monitoraggio di query (QMR) per limitare le query con tempi di esecuzione lunghi.

Utilizzando il dimensionamento simultaneo e il WLM automatico, puoi supportare un numero virtualmente illimitato di utenti simultanei e query simultanee, con prestazioni di query a velocità costante. Per ulteriori informazioni, consultare Utilizzo del dimensionamento della simultaneità.

> **Note**
>
> Consigliamo di creare un gruppo di parametri e di scegliere il WLM automatico per gestire le risorse delle query. Per informazioni dettagliate su come migrare da WLM manuale a WLM automatica, consultare Migrazione da WLM manuale a WLM automatico.

WLM manuale

In alternativa, puoi gestire le prestazioni del sistema e l'esperienza degli utenti modificando la configurazione WLM per creare code separate per le query a esecuzione prolungata e le query a esecuzione breve. In fase di runtime, puoi indirizzare le query a queste code in base ai gruppi di utenti o ai gruppi di query. È possibile abilitare la configurazione manuale tramite la console Amazon Redshift impostando WLM manuale. In questo modo, specifichi le code utilizzate per gestire le query e i valori dei campi Memory (Memoria) e Concurrency on main (Simultaneità su principale). Con la configurazione manuale, puoi configurare fino a otto code di query e impostare il numero di query che possono essere eseguite contemporaneamente in ciascuna di queste code.

Puoi impostare le regole per indirizzare le query alle specifiche code in base all'utente che esegue la query o alle etichette specificate. Puoi inoltre configurare la quantità di memoria allocata a ciascuna coda, in modo che le query di grandi dimensioni vengano eseguite nelle code con più memoria rispetto alle altre code. È inoltre possibile configurare una regola di monitoraggio di query (QMR) per limitare le query con tempi di esecuzione lunghi. Per ulteriori informazioni, consultare Implementazione di WLM manuale.

> **Note**
>
> Consigliamo di configurare le code di query WLM manuale con al massimo un totale di 15 slot di query. Per ulteriori informazioni, consulta Livello di simultaneità.

Limitazioni delle code WLM

Si noti che per quanto riguarda una configurazione WLM manuale, il numero massimo di slot che è possibile allocare a una coda è 50. Tuttavia, ciò non significa che in una configurazione WLM

automatica, un cluster di Amazon Redshift esegua sempre 50 query contemporaneamente. Questo può cambiare in base alle esigenze di memoria o ad altri tipi di allocazione delle risorse nel cluster.

Casi d'uso per Auto WLM e Manual WLM

Usa Auto WLM se desideri che Amazon Redshift gestisca il modo in cui le risorse sono divise per l'esecuzione simultanea di query. L'utilizzo di Auto WLM spesso comporta una velocità di trasmissione effettiva più elevata rispetto a Manual WLM. Con Auto WLM, puoi definire le priorità delle query per i carichi di lavoro in coda. Per ulteriori informazioni sulla priorità delle query, consulta Priorità delle query.

Usa Manual WLM quando desideri un maggiore controllo sulla simultaneità.

Argomenti

Modifica della configurazione WLM

Il modo più semplice per modificare la configurazione WLM consiste nell'usare la console Amazon Redshift. Puoi anche utilizzare l' AWS CLI API o Amazon Redshift.

Quando alterni le impostazioni del cluster tra WLM automatico e manuale, il cluster viene impostato sullo stato `pending reboot`. La modifica non ha effetto fino al riavvio successivo del cluster.

Per informazioni dettagliate sulla modifica delle configurazioni WLM, consulta Configurazione della gestione del carico di lavoro nella Guida alla gestione di Amazon Redshift.

Migrazione da WLM manuale a WLM automatico

Per massimizzare il troughput di sistema e utilizzare le risorse in modo più efficace, consigliamo la configurazione del WLM automatico per le code. Tieni a mente il seguente approccio per configurare una transizione senza ostacoli dal WLM manuale al WLM automatico.

Per effettuare la migrazione dal WLM manuale al WLM automatico e utilizzare le priorità di query, consigliamo di creare un nuovo gruppo di parametri e di collegarlo al cluster. Per ulteriori informazioni, consulta Gruppi di parametri di Amazon Redshift nella Guida alla gestione di Amazon Redshift.

> **Important**
>
> Se modifichi il gruppo di parametri o se passi dal WLM manuale al WLM automatico, è necessario riavviare il cluster. Per ulteriori informazioni, consultare Proprietà di configurazione dinamiche e statiche WLM.

Prendiamo un esempio in cui ci sono tre code con WLM manuale. Una coda ciascuna per un carico di lavoro ETL, un carico analitico e uno di data science. Il carico di lavoro ETL viene eseguito ogni 6 ore, quello analitico durante tutto il giorno e il carico di data science può avere picchi in qualsiasi momento. Con il WLM manuale, specifichi la memoria e la simultaneità di ogni coda del carico di lavoro in base all'importanza di ogni carico per il business. Specificare la memoria e la simultaneità non è solo difficile da immaginare, ma produce anche un partizionamento statico delle risorse del cluster, che vengono così sprecate se è in esecuzione solo un sottoinsieme dei carichi di lavoro.

Puoi utilizzare il WLM automatico con priorità di query per indicare le priorità relative dei carichi di lavoro, evitando i problemi precedenti. Per questo esempio, effettua la procedura riportata di seguito:

- Creare un nuovo parametro e passare alla modalità Auto WLM (WLM automatico).
- Aggiungere code per ognuno dei tre carichi di lavoro: ETL, analitico e di data science. Utilizzare gli stessi gruppi di utenti per ogni carico di lavoro che è stato utilizzato con modalità Manual WLM (WLM manuale).
- Impostare la priorità per il carico di lavoro ETL su `High`, del carico di lavoro analitico su `Normal` e di quello di data science su `Low`. Tali priorità rispecchiano le priorità aziendali dei diversi carichi di lavoro o gruppi di utenti.

- Opzionalmente, si può abilitare il dimensionamento simultaneo per la coda del carico di lavoro analitico o di data science in modo che le query in queste code raggiungano prestazioni coerenti anche quando il carico di lavoro ETL viene eseguito ogni 6 ore.

Con le priorità di query, quando solo il carico di lavoro analitico è in esecuzione nel cluster, riceve tutte le risorse. Ciò garantisce una velocità di trasmissione effettiva elevata con un migliore utilizzo del sistema. Tuttavia, quando si avvia il carico di lavoro ETL, questo ha la precedenza, dal momento che ha una priorità maggiore. Le query eseguite come parte del carico di lavoro ETL hanno la priorità durante l'ammissione oltre a un'allocazione preferenziale delle risorse dopo l'ammissione. Di conseguenza, il carico di lavoro ETL ottiene prestazioni prevedibili indipendentemente da qualsiasi altra operazione in esecuzione nel sistema. Le prestazioni prevedibili di un carico di lavoro con priorità elevata sono disponibili a scapito di altri carichi di lavoro con priorità minore che vengono eseguiti per un tempo più lungo, perché le query attendono il completamento di query più importanti. Oppure perché ricevono una frazione più piccola di risorse quando vengono eseguite simultaneamente a query con priorità maggiore. Gli algoritmi di pianificazione utilizzati da Amazon Redshift fanno in modo che le query con priorità minori non soffrano di un uso scarso delle risorse, ma che continuino ad avanzare, seppure a un ritmo più lento.

> **Note**
>
> - Il campo timeout non è disponibile nel WLM automatico. Utilizza invece la regola QMR, `query_execution_time`. Per ulteriori informazioni, consultare Regole di monitoraggio delle query WLM.
> - L'operazione QMR, HOP, non è applicabile al WLM automatico. Utilizza invece l'operazione `change priority`. Per ulteriori informazioni, consulta Regole di monitoraggio delle query WLM.
> - I cluster utilizzano le code WLM automatiche e manuali in modo diverso, il che può creare confusione con le configurazioni. Ad esempio, è possibile configurare la proprietà di priorità nelle code WLM automatiche ma non nelle code WLM manuali. Per questo motivo, all'interno di un gruppo di parametri è preferibile evitare il mix tra code WLM automatiche e code WLM manuali. Crea invece un nuovo gruppo di parametri durante la migrazione al WLM automatico.

Implementazione del WLM automatico

Con la gestione automatica del carico di lavoro (WLM), Amazon Redshift gestisce la simultaneità delle query e l'allocazione della memoria. Vengono create fino a otto code con identificatori della classe di servizio da 100 a 107. Ogni coda ha una priorità. Per ulteriori informazioni, consulta Priorità delle query.

Il WLM automatico determina la quantità di risorse necessaria alle query e regola la simultaneità in base al carico di lavoro. Quando nel sistema sono presenti query che richiedono quantità elevate di risorse (ad esempio, hash join tra tabelle di grandi dimensioni), il livello di simultaneità è inferiore. Se vengono inviate query più piccole (come inserimenti, eliminazioni, scansioni o semplici aggregazioni), il livello di simultaneità è maggiore.

WLM automatico è distinto dall'accelerazione di query brevi (short query acceleration, SQA) e valuta le query in modo diverso. WLM automatico e SQA lavorano insieme per consentire il completamento di query leggere e di breve esecuzione anche in presenza di code lunghe e che fanno un uso intensivo delle risorse. Per ulteriori informazioni su SQA, consultare Utilizzo dall'accelerazione di query brevi.

Amazon Redshift abilita la WLM automatica tramite gruppi di parametri:

- Se i cluster utilizzano il gruppo di parametri di default, Amazon Redshift abilita la WLM automatica.
- Se i cluster utilizzano gruppi di parametri personalizzati, puoi configurarli per l'abilitazione di WLM automatico. Ti consigliamo di creare un gruppo di parametri distinto per la configurazione di WLM automatico.

Per configurare WLM, modifica il parametro `wlm_json_configuration` in un gruppo di parametri che può essere associato a uno o più cluster. Per ulteriori informazioni, consultare Modifica della configurazione WLM.

Definisci le code di query all'interno della configurazione WLM. Puoi aggiungere altre code di query alla configurazione WLM predefinita, fino a un totale di otto code dell'utente. Per ogni coda di query puoi configurare le seguenti opzioni:

- Priority (Priorità)
- Modalità dimensionamento simultaneo
- Gruppi di utenti

- Gruppi di query
- Regole di monitoraggio delle query

Priority (Priorità)

Puoi definire l'importanza relativa delle query in un carico di lavoro impostando un valore di priorità. La priorità viene specificata per una coda e la ereditano tutte le query associate alla coda. Per ulteriori informazioni, consultare Priorità delle query.

Modalità dimensionamento simultaneo

Quando il dimensionamento simultaneo è abilitato, Amazon Redshift aggiunge automaticamente ulteriore capacità del cluster quando necessario per elaborare un aumento delle query di lettura r scrittura simultanee. Gli utenti visualizzano sempre i dati più recenti, indipendentemente dal fatto che le query vengano eseguite nel cluster principale o in un cluster di dimensionamento simultaneo.

È possibile gestire le query inviate al cluster di dimensionamento simultaneo configurando le code di gestione del carico di lavoro. Quando abiliti il dimensionamento simultaneo per una coda, le query idonee vengono inviate al cluster di dimensionamento simultaneo anziché attendere in coda. Per ulteriori informazioni, consulta Utilizzo del dimensionamento della simultaneità.

Gruppi di utenti

Puoi assegnare un set di gruppi di utenti a una coda specificando il nome di ogni gruppo di utenti o utilizzando i caratteri jolly. Quando un membro di un gruppo utenti elencato esegue una query, quella query viene eseguita nella coda corrispondente. Non esiste un limite impostato per il numero di gruppi di utenti che possono essere assegnati a una coda. Per ulteriori informazioni, consultare Assegnazione delle query alle code in base ai gruppi di utenti.

Gruppi di query

Puoi assegnare un set di gruppi di query a una coda specificando il nome di ogni gruppo di query o utilizzando i caratteri jolly. Un gruppo di query è semplicemente un'etichetta. In fase di runtime, puoi assegnare l'etichetta del gruppo di query a una serie di query. Qualsiasi query che viene assegnata a un gruppo di query elencato verrà eseguita nella coda corrispondente. Non esiste un limite definito per il numero di gruppi di query che possono essere assegnati a una coda. Per ulteriori informazioni, consulta Assegnazione di una query a un gruppo di utenti.

Caratteri jolly

Se i caratteri jolly sono abilitati nella configurazione della coda WLM, è possibile assegnare gruppi di utenti e gruppi di query a una coda individualmente o utilizzando i caratteri jolly nello stile shell Unix. La corrispondenza del modello fa distinzione tra maiuscole e minuscole.

Ad esempio, il carattere jolly “*” corrisponde a qualsiasi numero di caratteri. Pertanto, se aggiungi `dba_*` all'elenco dei gruppi di utenti di una coda, qualsiasi query eseguita dagli utenti che appartiene a un gruppo con un nome che inizia con `dba_` verrà assegnata a quella coda. Alcuni esempi sono `dba_admin` o `DBA_primary`. Il carattere jolly “?” corrisponde a qualsiasi carattere singolo. Pertanto, se la coda include il gruppo di utenti `dba?1`, i gruppi di utenti denominati `dba11` e `dba21` corrisponderanno, ma non quelli chiamati `dba12`.

Per impostazione predefinita, i caratteri jolly non sono abilitati.

Regole di monitoraggio delle query

Le regole di monitoraggio delle query definiscono i limiti delle prestazioni basati sui parametri per le code WLM e specificano l'azione da intraprendere quando una query oltrepassa tali limiti. Ad esempio, per una coda dedicata alle query di breve durata, puoi creare una regola che annulla le query eseguite per più di 60 secondi. Per tracciare le query mal progettate, potresti avere un'altra regola che registra le query che contengono cicli annidati. Per ulteriori informazioni, consultare Regole di monitoraggio delle query WLM.

Controllo di WLM automatico

Per controllare se WLM automatico è abilitato, esegui la seguente query. Se la query restituisce almeno una riga, WLM automatico è abilitato.

```
select * from stv_wlm_service_class_config
where service_class >= 100;
```

La seguente query mostra il numero di query che hanno sostato in ciascuna coda di query (classe di servizio). Mostra anche il tempo medio di esecuzione, il numero di query con tempo di attesa al 90° percentile e il tempo medio di attesa. Le query di WLM automatico utilizzano le classi di servizio da 100 a 107.

```
select final_state, service_class, count(*), avg(total_exec_time),
```

```
percentile_cont(0.9) within group (order by total_queue_time), avg(total_queue_time)
from stl_wlm_query where userid >= 100 group by 1,2 order by 2,1;
```

Per sapere quali query sono state eseguite e completate correttamente da WLM automatico, esegui la seguente query.

```
select a.queue_start_time, a.total_exec_time, label, trim(querytxt)
from stl_wlm_query a, stl_query b
where a.query = b.query and a.service_class >= 100 and a.final_state = 'Completed'
order by b.query desc limit 5;
```

Priorità delle query

Non tutte le query hanno la stessa importanza; spesso le prestazioni di un carico di lavoro o di un set di utenti possono essere più importanti. Se hai abilitato WLM automatico, puoi definire l'importanza relativa delle query in un carico di lavoro impostando un valore di priorità. La priorità viene specificata per una coda e la ereditano tutte le query associate alla coda. Le query vengono associate a una coda mappando gruppi di utenti e gruppi di query alla coda. Puoi impostare le seguenti priorità (elencate da quella più alta a quella più bassa):

1. `HIGHEST`
2. `HIGH`
3. `NORMAL`
4. `LOW`
5. `LOWEST`

Gli amministratori utilizzano queste priorità per illustrare l'importanza relativa dei carichi di lavoro quando query con priorità diverse si contendono le stesse risorse. Amazon Redshift utilizza la priorità quando lascia query nel sistema e per determinare la quantità di risorse allocate a una query. Per impostazione predefinita, le query vengono eseguite con priorità impostata su `NORMAL`.

Una priorità aggiuntiva, `CRITICAL`, superiore a `HIGHEST`, è disponibile per gli utenti con privilegi avanzati. Per impostare questa priorità, puoi utilizzare le funzioni CHANGE_QUERY_PRIORITY, CHANGE_SESSION_PRIORITY e CHANGE_USER_PRIORITY. Per concedere a un utente di database l'autorizzazione per l'uso di queste funzioni, puoi creare una procedura archiviata e concedere l'autorizzazione a un utente. Per un esempio, consultare CHANGE_SESSION_PRIORITY.

> **Note**
>
> Si può eseguire solo una query `CRITICAL` alla volta.

Prendiamo un esempio in cui la priorità di un carico di lavoro ETL (estrazione, trasformazione, carico) è superiore a quella del carico di lavoro analitico. Il carico di lavoro ETL viene eseguito ogni sei ore, mentre quello analitico è in esecuzione per tutto il giorno. Quando solo il carico di lavoro analitico è in esecuzione sul cluster, attira tutte le risorse su di sé, ottenendo un elevato throughput e un utilizzo ottimale del sistema. Tuttavia, quando si avvia il carico di lavoro ETL, questo ha la precedenza, dal momento che ha una priorità maggiore. Le query eseguite come parte del carico di lavoro ETL hanno la precedenza durante l'ammissione e un'allocazione preferenziale delle risorse dopo l'ammissione. Di conseguenza, il carico di lavoro ETL ottiene prestazioni prevedibili indipendentemente da qualsiasi altra operazione in esecuzione nel sistema. In questo modo, offre prestazioni prevedibili e la possibilità per gli amministratori di fornire contratti sul livello di servizio (service level agreements, SLA) agli utenti aziendali.

All'interno di un determinato cluster, le prestazioni prevedibili di un carico di lavoro con priorità superiori vanno a scapito di altri carichi di lavoro con priorità minori. L'esecuzione dei carichi di lavoro con priorità inferiori può durare più a lungo perché le loro query attendono il completamento di query più importanti. O perché ricevono una frazione più piccola di risorse quando vengono eseguite simultaneamente a query con priorità maggiore. Le query con priorità minore non soffrono un uso eccessivo delle risorse, ma progrediscono a un ritmo più lento.

Nell'esempio precedente, l'amministratore può abilitare il dimensionamento della simultaneità per il carico di lavoro analitico. In questo modo il carico di lavoro mantiene il proprio throughput anche se il carico di lavoro ETL viene eseguito con priorità elevata.

Configurazione della priorità di coda

Se hai abilitato WLM automatico, ogni coda ha un valore di priorità. Le query vengono instradata in base ai gruppi di utenti e ai gruppi di query. Iniziare con una priorità della coda impostata su `NORMAL`. Imposta la priorità come maggiore o minore in base al carico di lavoro associato ai gruppi di utenti e ai gruppi di query della coda.

È possibile modificare la priorità di una coda nella console Amazon Redshift. Nella console Amazon Redshift, la pagina Gestione del carico di lavoro visualizza le code e permette la modifica delle proprietà come Priorità. Per impostare la priorità tramite la CLI o le operazioni API; utilizza il

parametro `wlm_json_configuration`. Per informazioni, consulta Configurazione della gestione del carico di lavoro nella Guida alla gestione di Amazon Redshift.

Il seguente esempio `wlm_json_configuration` definisce tre gruppi di utenti (`ingest`, `reporting` e `analytics`). Le query inviate dagli utenti di uno di questi gruppi vengono eseguite rispettivamente con priorità `highest`, `normal` e `low`.

```
[
    {
        "user_group": [
            "ingest"
        ],
        "priority": "highest",
        "queue_type": "auto"
    },
    {
        "user_group": [
            "reporting"
        ],
        "priority": "normal",
        "queue_type": "auto"
    },
    {
        "user_group": [
            "analytics"
        ],
        "priority": "low",
        "queue_type": "auto",
        "auto_wlm": true
    }
]
```

Modifica della priorità delle query con le regole di monitoraggio delle query

Le regole di monitoraggio delle query (QMR) ti permettono di modificare la priorità di una query in base al suo comportamento durante l'esecuzione. Ciò avviene quando, oltre a un'operazione, specifichi l'attributo di priorità in un predicato QMR. Per ulteriori informazioni, consulta Regole di monitoraggio delle query WLM.

Ad esempio, puoi definire una regola per annullare tutte le query classificate con priorità `high` che vengono eseguite per più di 10 minuti.

```
"rules" :[
  {
    "rule_name":"rule_abort",
    "predicate":[
      {
        "metric_name":"query_cpu_time",
        "operator":">",
        "value":600
      },
      {
        "metric_name":"query_priority",
        "operator":"=",
        "value":"high"
      }
    ],
    "action":"abort"
  }
]
```

Un altro esempio è quello di definite una regola che modifica la priorità in `lowest` per tutte le query con priorità attuale `normal` che riversano su disco più di 1 TB.

```
"rules":[
  {
    "rule_name":"rule_change_priority",
    "predicate":[
      {
        "metric_name":"query_temp_blocks_to_disk",
        "operator":">",
        "value":1000000
      },
      {
        "metric_name":"query_priority",
        "operator":"=",
        "value":"normal"
      }
    ],
    "action":"change_query_priority",
    "value":"lowest"
  }
]
```

Monitoraggio della priorità delle query

Per visualizzare la priorità delle query in esecuzione e in attesa, osserva la colonna `query_priority` nella tabella di sistema stv_wlm_query_state.

```
query    | service_cl | wlm_start_time             | state            | queue_time |
 query_priority
---------+------------+----------------------------+------------------+------------
+----------------
2673299  | 102        | 2019-06-24 17:35:38.866356 | QueuedWaiting    | 265116     |
 Highest
2673236  | 101        | 2019-06-24 17:35:33.313854 | Running          | 0          |
 Highest
2673265  | 102        | 2019-06-24 17:35:33.523332 | Running          | 0          |
 High
2673284  | 102        | 2019-06-24 17:35:38.477366 | Running          | 0          |
 Highest
2673288  | 102        | 2019-06-24 17:35:38.621819 | Running          | 0          |
 Highest
2673310  | 103        | 2019-06-24 17:35:39.068513 | QueuedWaiting    | 62970      |
 High
2673303  | 102        | 2019-06-24 17:35:38.968921 | QueuedWaiting    | 162560     |
 Normal
2673306  | 104        | 2019-06-24 17:35:39.002733 | QueuedWaiting    | 128691     |
 Lowest
```

Per un elenco della priorità delle query completate, visualizza la colonna `query_priority` nella tabella di sistema stl_wlm_query.

```
select query, service_class as svclass, service_class_start_time as starttime,
 query_priority
from stl_wlm_query order by 3 desc limit 10;
```

```
  query  | svclass |         starttime          |    query_priority
---------+---------+----------------------------+----------------------
 2723254 |     100 | 2019-06-24 18:14:50.780094 | Normal
 2723251 |     102 | 2019-06-24 18:14:50.749961 | Highest
 2723246 |     102 | 2019-06-24 18:14:50.725275 | Highest
 2723244 |     103 | 2019-06-24 18:14:50.719241 | High
 2723243 |     101 | 2019-06-24 18:14:50.699325 | Low
```

```
2723242 |     102 | 2019-06-24 18:14:50.692573 | Highest
2723239 |     101 | 2019-06-24 18:14:50.668535 | Low
2723237 |     102 | 2019-06-24 18:14:50.661918 | Highest
2723236 |     102 | 2019-06-24 18:14:50.643636 | Highest
```

Per ottimizzare la velocità effettiva del carico di lavoro, Amazon Redshift potrebbe modificare la priorità delle query inviate dall'utente. Amazon Redshift utilizza algoritmi avanzati di machine learning per determinare quando questa ottimizzazione avvantaggia il carico di lavoro e la applica automaticamente quando vengono soddisfatte tutte le seguenti condizioni.

- La gestione automatica del carico di lavoro (WLM) è abilitata.
- È definita una sola coda WLM.
- Non sono state definite regole di monitoraggio delle query (QMR) che impostano la priorità della query. Tali regole includono il parametro QMR `query_priority` o l'operazione QMR `change_query_priority`. Per ulteriori informazioni, consultare Regole di monitoraggio delle query WLM.

Implementazione di WLM manuale

Con il WLM manuale, è possibile gestire le prestazioni del sistema e l'esperienza degli utenti modificando la configurazione WLM per creare code separate per le query a esecuzione prolungata e le query a esecuzione breve.

Quando gli utenti eseguono query in Amazon Redshift, le query vengono indirizzate alle code di query. Ogni coda di query contiene un numero di slot di query. A ogni coda viene assegnata una parte della memoria disponibile del cluster. La memoria di una coda viene ripartita tra gli slot di query della coda. È possibile abilitare Amazon Redshift in modo che gestisca la simultaneità delle query con la WLM automatica. Per ulteriori informazioni, consultare Implementazione del WLM automatico.

Oppure puoi configurare proprietà WLM per ogni coda di query. Tale operazione serve per specificare il modo in cui la memoria viene allocata agli slot e come le query vengono indirizzate alle code specifiche durante il runtime. Inoltre, puoi configurare le proprietà WLM per annullare le query di lunga esecuzione.

Per impostazione predefinita, Amazon Redshift configura le code di query elencate di seguito:

- Una coda dell'utente con privilegi avanzati

La coda dell'utente con privilegi avanzati è riservata solo agli utenti con privilegi avanzati e non può essere configurata. Utilizzare questa coda solo se è necessario eseguire query che interessano il sistema o per la risoluzione dei problemi. Ad esempio, utilizzare questa coda quando è necessario annullare la query a esecuzione prolungata di un utente o aggiungere utenti al database. Non utilizzarla per eseguire le query di routine. La coda non viene mostrata nella console, ma è presente nelle tabelle di sistema del database come quinta coda. Per eseguire una query nella coda dell'utente con privilegi avanzati, un utente deve aver eseguito l'accesso come utente con privilegi avanzati e deve eseguire la query utilizzando il gruppo di query predefinito `superuser`.

- Una coda dell'utente predefinita

  La coda predefinita è inizialmente configurata per eseguire cinque query contemporaneamente. Quando si utilizza il WLM manuale, è possibile modificare le proprietà di simultaneità, timeout e allocazione della memoria per la coda predefinita, ma non è possibile specificare gruppi di utenti o gruppi di query. La coda predefinita deve essere l'ultima coda nella configurazione WLM. Qualsiasi query che non viene instradata ad altre code, viene eseguita nella coda predefinita.

Le code di query sono definite nella configurazione WLM. La configurazione WLM è un parametro modificabile (`wlm_json_configuration`) in un gruppo di parametri che può essere associato a uno o più cluster. Per informazioni, consulta Configurazione della gestione del carico di lavoro nella Guida alla gestione di Amazon Redshift.

Puoi aggiungere altre code di query alla configurazione WLM predefinita, fino a un totale di otto code dell'utente. Per ogni coda di query puoi configurare le seguenti opzioni:

- Modalità dimensionamento simultaneo
- Livello di simultaneità
- Gruppi di utenti
- Gruppi di query
- Percentuale di memoria WLM da utilizzare
- Timeout WLM
- Hop della coda di query WLM
- Regole di monitoraggio delle query

Modalità dimensionamento simultaneo

Quando il dimensionamento simultaneo è abilitato, Amazon Redshift aggiunge automaticamente ulteriore capacità del cluster quando necessario per elaborare un aumento delle query di lettura r scrittura simultanee. Gli utenti visualizzano sempre i dati più recenti, indipendentemente dal fatto che le query vengano eseguite nel cluster principale o in un cluster di dimensionamento simultaneo.

È possibile gestire le query inviate al cluster di dimensionamento simultaneo configurando le code di gestione del carico di lavoro. Quando abiliti il dimensionamento simultaneo per una coda, le query idonee vengono inviate al cluster di dimensionamento simultaneo anziché attendere in coda. Per ulteriori informazioni, consulta Utilizzo del dimensionamento della simultaneità.

Livello di simultaneità

Le query in una coda vengono eseguite contemporaneamente fino al raggiungimento del conteggio degli slot di query WLM, detto livello di simultaneità, definito per la coda. Le query successive quindi attendono in coda.

> **Note**
>
> Il livello di simultaneità WLM è diverso dal numero di connessioni dell'utente simultanee che possono essere eseguite a un cluster. Per ulteriori informazioni, consulta Connessione a un cluster nella Guida alla gestione di Amazon Redshift.

In una configurazione WLM automatica (consigliata), il livello di simultaneità è impostato su Auto. Amazon Redshift alloca dinamicamente la memoria alle query, che successivamente determina quante se ne devono eseguire contemporaneamente. Ciò è basato sulle risorse necessarie sia per le query esecuzione che per quelle in coda. Il WLM automatico non è configurabile. Per ulteriori informazioni, consulta Implementazione del WLM automatico.

In una configurazione WLM manuale, Amazon Redshift alloca in modo statico una quantità fissa di memoria a ciascuna coda. La memoria della coda è suddivisa equamente tra gli slot di query. Ad esempio, se a una coda è allocato il 20% della memoria di un cluster e ha 10 slot, a ciascuna query viene allocato il 2% della memoria del cluster. L'allocazione della memoria rimane fissa indipendentemente dal numero di query eseguite contemporaneamente. A causa di questa allocazione fissa di memoria, le query eseguite interamente in memoria quando il numero di slot è 5 potrebbe dover scrivere i risultati intermedi sul disco se il numero degli slot è aumentato a 20. In

questo caso, ogni condivisione della memoria della coda viene ridotta da 1/5 a 1/20. L'I/O del disco aggiuntivo potrebbe compromettere le prestazioni.

Il numero massimo degli slot per tutte le code definite dall'utente è 50. Questo limita gli slot totali per tutte le code, inclusa la coda predefinita. L'unica coda che non è soggetta al limite è quella riservata agli utenti con privilegi avanzati.

Per impostazione predefinita, le code WLM hanno un livello di simultaneità pari a 5. Il carico di lavoro potrebbe beneficiare di un livello di simultaneità più elevato in alcuni casi, come ad esempio il seguente:

- Se molte piccole query sono costrette ad attendere le query a esecuzione prolungata, crea una coda separata con un numero di slot superiore e assegna le query più piccole a quella coda. Una coda con un livello di simultaneità più alto ha meno memoria allocata per ogni spazio di query, ma le query più piccole richiedono meno memoria.

  > **Note**
  >
  > Se si abilita l'accelerazione di query brevi (SQA), WLM assegna automaticamente la priorità alle query brevi su query più lunghe, quindi non è necessaria una coda separata per le query brevi per la maggior parte dei flussi di lavoro. Per ulteriori informazioni, consulta Utilizzo dall'accelerazione di query brevi.

- Se sono presenti più query, ognuna delle quali accede ai dati di una singola sezione, impostare una coda WLM separata in modo da eseguire queste query contemporaneamente. Amazon Redshift assegna query simultanee a sezioni separate, consentendo l'esecuzione di più query in parallelo su più sezioni. Ad esempio, se una query è una semplice aggregazione con un predicato sulla chiave di distribuzione, i dati per la query si troveranno su una singola sezione.

Un esempio di WLM manuale

Questo esempio è uno scenario semplice di WLM manuale per mostrare come è possibile allocare slot e memoria. Implementa il WLM manuale con tre code, che sono le seguenti:

- Coda di importazione dei dati: è impostata per l'importazione dei dati. È allocato il 20% della memoria del cluster e dispone di 5 slot. Pertanto, possono essere eseguite contemporaneamente 5 query in coda e per ciascuna viene allocato il 4% della memoria.

- Coda data-scientist: progettata per query che richiedono un uso intensivo della memoria. È allocato il 40% della memoria del cluster e dispone di 5 slot. Pertanto, possono essere eseguite contemporaneamente 5 query e per ciascuna viene allocato l'8% della memoria.
- coda predefinita: è progettata per la maggior parte degli utenti dell'organizzazione. Ciò include i gruppi di vendita e contabilità che in genere eseguono query di breve o media esecuzione e che non sono complicate. È allocato il 40% della memoria del cluster e dispone di 40 slot. 40 query possono essere eseguite contemporaneamente in questa coda, e a ogni query è allocato l'1% della memoria. Questo è il numero massimo di slot che possono essere allocati per questa coda perché tra tutte le code il limite è 50.

Se è esecuzione il WLM automatico e il carico di lavoro richiede l'esecuzione in parallelo di 15 query, è consigliabile abilitare il dimensionamento simultaneo. Questo perché l'aumento del numero di slot di query oltre 15 potrebbe creare conflitti per le risorse di sistema e limitare la velocità di trasmissione effettiva complessiva di un singolo cluster. Con il dimensionamento simultaneo, puoi eseguire centinaia di query in parallelo fino a un numero configurato di cluster con dimensionamento simultaneo. Il numero di cluster con dimensionamento simultaneo che è possibile utilizzare è controllato da max_concurrency_scaling_clusters. Per ulteriori informazioni sul dimensionamento simultaneo, consultare Utilizzo del dimensionamento della simultaneità.

Per ulteriori informazioni, consulta Miglioramento delle prestazioni della query di .

Gruppi di utenti

Puoi assegnare un set di gruppi di utenti a una coda specificando il nome di ogni gruppo di utenti o utilizzando i caratteri jolly. Quando un membro di un gruppo utenti elencato esegue una query, quella query viene eseguita nella coda corrispondente. Non esiste un limite impostato per il numero di gruppi di utenti che possono essere assegnati a una coda. Per ulteriori informazioni, consultare Assegnazione delle query alle code in base ai gruppi di utenti.

Gruppi di query

Puoi assegnare un set di gruppi di query a una coda specificando il nome di ogni gruppo di query o utilizzando i caratteri jolly. Un gruppo di query è semplicemente un'etichetta. In fase di runtime, puoi assegnare l'etichetta del gruppo di query a una serie di query. Qualsiasi query che viene assegnata a un gruppo di query elencato verrà eseguita nella coda corrispondente. Non esiste un limite definito per il numero di gruppi di query che possono essere assegnati a una coda. Per ulteriori informazioni, consulta Assegnazione di una query a un gruppo di utenti.

Caratteri jolly

Se i caratteri jolly sono abilitati nella configurazione della coda WLM, puoi assegnare gruppi di utenti e gruppi di query a una coda individualmente o utilizzando i caratteri jolly nello stile shell Unix. La corrispondenza del modello fa distinzione tra maiuscole e minuscole.

Ad esempio, il carattere jolly “*” corrisponde a qualsiasi numero di caratteri. Pertanto, se aggiungi `dba_*` all'elenco dei gruppi di utenti di una coda, qualsiasi query eseguita dagli utenti che appartiene a un gruppo con un nome che inizia con `dba_` verrà assegnata a quella coda. Un paio di esempi sono `dba_admin` o `DBA_primary`. Il carattere jolly “?” corrisponde a qualsiasi carattere singolo. Pertanto, se la coda include il gruppo di utenti `dba?1`, i gruppi di utenti denominati `dba11` e `dba21` corrisponderanno, ma non quelli chiamati `dba12`.

I caratteri jolly sono disattivati per impostazione predefinita.

Percentuale di memoria WLM da utilizzare

Nella configurazione WLM automatica, la percentuale di memoria è impostata su **`auto`**. Per ulteriori informazioni, consultare Implementazione del WLM automatico.

In una configurazione WLM manuale, per specificare la quantità di memoria disponibile allocata a una query puoi impostare il parametro `WLM Memory Percent to Use`. Per impostazione predefinita, a ogni coda definita dall'utente viene allocata una porzione uguale della memoria disponibile per le query definite dall'utente. Ad esempio, se hai quattro code definite dall'utente, a ciascuna coda viene assegnato il 25 percento della memoria disponibile. La coda dell'utente con privilegi avanzati ha una sua memoria allocata che non può essere modificata. Per modificare l'allocazione, assegni una percentuale intera di memoria a ciascuna coda, fino a un totale del 100 percento. Qualsiasi memoria non allocata è gestita da Amazon Redshift e può essere temporaneamente assegnata a una coda, se la coda richiede memoria aggiuntiva per l'elaborazione.

Ad esempio, se configuri quattro code, puoi allocare la memoria come segue: 20 percento, 30 percento, 15 percento, 15 percento. Il restante 20 percento è non assegnato ed è gestito dal servizio.

Timeout WLM

Il timeout WLM (`max_execution_time`) è sconsigliato. Al contrario, è necessario creare una regola di monitoraggio di query (QMR) utilizzando `query_execution_time` per limitare il tempo di esecuzione trascorso per una query. Per ulteriori informazioni, consultare Regole di monitoraggio delle query WLM.

Per limitare il tempo di utilizzo delle query in una determinata coda WLM, puoi impostare il valore di timeout WLM per ogni coda. Il parametro di timeout specifica la quantità di tempo, in millisecondi, che Amazon Redshift attende per l'esecuzione di una query prima di annullare o saltare la query. Il timeout si basa sul tempo di esecuzione della query e non include il tempo trascorso in attesa in una coda.

WLM tenta di saltare le istruzioni CREATE TABLE AS (CTAS) e le query di sola lettura, ad esempio le istruzioni SELECT. Le query che non possono essere saltate vengono annullate. Per ulteriori informazioni, consultare Hop della coda di query WLM.

Il timeout WLM non si applica a una query il cui stato è returning. Per visualizzare lo stato di una query, consultare la tabella di sistema STV_WLM_QUERY_STATE. Le istruzioni COPY e le operazioni di manutenzione, come ANALYZE e VACUUM, non sono soggette al timeout WLM.

La funzione del timeout WLM è simile al parametro di configurazione statement_timeout, con la differenza che quando il parametro di configurazione `statement_timeout` si applica all'intero cluster, il timeout WLM è specifico di una singola coda nella configurazione WLM.

Se si specifica anche statement_timeout, viene utilizzato il valore più basso di statement_timeout e il timeout WLM (max_execution_time).

Regole di monitoraggio delle query

Le regole di monitoraggio delle query definiscono i limiti delle prestazioni basati sui parametri per le code WLM e specificano l'azione da intraprendere quando una query oltrepassa tali limiti. Ad esempio, per una coda dedicata alle query di breve durata, puoi creare una regola che annulla le query eseguite per più di 60 secondi. Per tracciare le query mal progettate, potresti avere un'altra regola che registra le query che contengono cicli annidati. Per ulteriori informazioni, consultare Regole di monitoraggio delle query WLM.

Hop della coda di query WLM

Una query può essere saltata a causa di un timeout WLM o un'operazione hop (QMR) di monitoraggio della query. Puoi saltare le query solo in una configurazione WLM manuale.

Quando una query viene saltata, WLM tenta di indirizzare la query alla successiva coda corrispondente in base alle regole di assegnazione delle code WLM. Se la query non corrisponde ad alcuna altra definizione di coda, la query viene annullata. Non viene assegnata alla coda predefinita.

Operazioni relative al timeout WLM

La tabella seguente riepiloga il comportamento dei diversi tipi di query con un timeout WLM.

Tipo di query	Azione
INSERT, UPDATE e DELETE	Annulla
Funzioni definite dall'utente (FDU)	Annulla
UNLOAD	Annulla
COPY	Continua l'esecuzione
Operazioni di manutenzione	Continua l'esecuzione
Query di sola lettura nello stato `returning`	Continua l'esecuzione
Query di sola lettura nello stato `running`	Viene riassegnata o riavviata
CREATE TABLE AS (CTAS), SELECT INTO	Viene riassegnata o riavviata

Hop della coda di timeout WLM

WLM salta i seguenti tipi di query quando generano un timeout:

- Query di sola lettura, come le istruzioni SELECT, che si trovano nello stato WLM `running`. Per trovare lo stato WLM di una query, consultare la colonna STATE della tabella di sistema STV_WLM_QUERY_STATE.
- Istruzioni CREATE TABLE AS (CTAS). L'hop della coda WLM supporta le istruzioni CTAS definite dall'utente e generate dal sistema.
- Istruzioni SELECT INTO.

Le query che non sono soggette al timeout WLM continuano a essere eseguite nella coda originale fino al completamento. I seguenti tipi di query non sono soggetti al timeout WLM:

- Istruzioni COPY
- Operazioni di manutenzione come ANALYZE e VACUUM

- Query di sola lettura, come le istruzioni SELECT, che sono nello stato WLM `returning`. Per trovare lo stato WLM di una query, consultare la colonna STATE della tabella di sistema STV_WLM_QUERY_STATE.

Le query che non sono idonee per l'hop da un timeout WLM vengono annullate quando generano il timeout. I seguenti tipi di query non sono idonei per l'hop da un timeout WLM:

- Istruzioni INSERT, UPDATE e DELETE
- Istruzioni UNLOAD
- Funzioni definite dall'utente (FDU)

Query riavviate e riassegnate per timeout WLM

Quando una query viene saltata e non viene trovata alcuna coda corrispondente, la query viene annullata.

Quando una query viene saltata e viene trovata una coda corrispondente, WLM tenta di riassegnare la query alla nuova coda. Se una query non può essere riassegnata, viene riavviata nella nuova coda, come descritto di seguito.

Una query viene riassegnata solo se tutte le seguenti considerazioni sono vere:

- Viene trovata una coda corrispondente.
- La nuova coda ha abbastanza slot liberi per eseguire la query. Una query potrebbe richiedere più slot se il parametro wlm_query_slot_count è stato impostato su un valore superiore a 1.
- La nuova coda ha almeno la stessa memoria disponibile utilizzata attualmente dalla query.

Se la query viene riassegnata, la query continua a essere eseguita nella nuova coda. I risultati intermedi vengono conservati, per cui l'effetto sul tempo totale di esecuzione è minimo.

Se la query non può essere riassegnata, la query viene annullata e riavviata nella nuova coda. I risultati intermedi sono eliminati. La query attende nella coda, quindi inizia a essere eseguita quando sono disponibili sufficienti slot.

Operazioni di hop QMR

La tabella seguente riepiloga il comportamento dei diversi tipi di query con un'operazione hop QMR.

Tipo di query	Azione
COPY	Continua l'esecuzione
Operazioni di manutenzione	Continua l'esecuzione
Funzioni definite dall'utente (FDU)	Continua l'esecuzione
UNLOAD	Viene riassegnata o continua l'esecuzione
INSERT, UPDATE e DELETE	Viene riassegnata o continua l'esecuzione
Query di sola lettura nello stato `returning`	Viene riassegnata o continua l'esecuzione
Query di sola lettura nello stato `running`	Viene riassegnata o riavviata
CREATE TABLE AS (CTAS), SELECT INTO	Viene riassegnata o riavviata

Per scoprire se una query che è stata saltata da QMR è stata riassegnata, riavviata o annullata, eseguire una query sulla tabella di log del sistema STL_WLM_RULE_ACTION.

Query riavviate e riassegnate per operazione hop QMR

Quando una query viene saltata e non viene trovata alcuna coda corrispondente, la query viene annullata.

Quando una query viene saltata e viene trovata una coda corrispondente, WLM tenta di riassegnare la query alla nuova coda. Se una query non può essere riassegnata, viene riavviata nella nuova coda o continua l'esecuzione nella coda originale, come descritto di seguito.

Una query viene riassegnata solo se tutte le seguenti considerazioni sono vere:

- Viene trovata una coda corrispondente.
- La nuova coda ha abbastanza slot liberi per eseguire la query. Una query potrebbe richiedere più slot se il parametro wlm_query_slot_count è stato impostato su un valore superiore a 1.
- La nuova coda ha almeno la stessa memoria disponibile utilizzata attualmente dalla query.

Se la query viene riassegnata, la query continua a essere eseguita nella nuova coda. I risultati intermedi vengono conservati, per cui l'effetto sul tempo totale di esecuzione è minimo.

Se una query non può essere riassegnata, la query viene riavviata o continua l'esecuzione nella coda originale. Se la query viene riavviata, la query viene annullata e riavviata nella nuova coda. I risultati intermedi sono eliminati. La query attende nella coda, quindi inizia l'esecuzione quando sono disponibili sufficienti slot.

Tutorial: Configurazione delle code di gestione manuale del carico di lavoro (WLM)

Panoramica

Consigliamo di configurare la gestione automatica del carico di lavoro (WLM) in Amazon Redshift. Per ulteriori informazioni sulla funzionalità WLM automatica, consultare Implementazione della gestione del carico di lavoro. Tuttavia, se c'è bisogno di più code WLM, questo tutorial introdurrà al processo di configurazione della gestione manuale del carico di lavoro (WLM) in Amazon Redshift. Grazie alla configurazione WLM manuale, puoi migliorare le prestazioni relative alle query e l'assegnazione delle risorse nel cluster.

Amazon Redshift instrada le query degli utenti alle code per l'elaborazione. Le modalità con cui le query devono essere instradate alle code vengono definite da WLM . Per impostazione predefinita, in Amazon Redshift sono disponibili due code per le query: una per gli utenti con privilegi avanzati e una per gli utenti. La coda dell'utente con privilegi avanzati non può essere configurata e può elaborare una sola query per volta. È consigliabile riservare questa coda solo per la risoluzione dei problemi. La coda per gli utenti può elaborare fino a cinque code per volta tuttavia, se necessario, si può modificare il livello di simultaneità della coda.

Se più utenti eseguono query sul database, potrebbe essere più efficace una configurazione diversa. Ad esempio, se alcuni utenti eseguono operazioni che richiedono un numero elevato di risorse, come VACUUM, queste potrebbero avere un impatto negativo sulle query meno impegnative, ad esempio i report. Potresti valutare l'opportunità di aggiungere altre code e di configurarle per carichi di lavoro diversi.

Tempo previsto: 75 minuti

Costo previsto: 50 centesimi

Prerequisiti

Sono necessari un cluster Amazon Redshift, il database TICKIT di esempio e lo strumento client Amazon Redshift RSQL. Se non sono ancora configurati, consulta la Guida alle operazioni di base di Amazon Redshift e Amazon Redshift RSQL.

Sections

Sezione 1: informazioni sul comportamento predefinito di elaborazione delle code

Prima di iniziare a configurare la WLM manuale, può essere utile conoscere il comportamento di default dell'elaborazione delle code in Amazon Redshift. In questa sezione creerai due viste database che restituiscono informazioni da diverse tabelle di sistema. Quindi eseguirai alcune query di prova per vedere come vengono instradate per impostazione predefinita. Per ulteriori informazioni sulle tabelle di sistema, consultare Riferimento di tabelle e viste di sistema.

Fase 1: creazione della vista WLM_QUEUE_STATE_VW

In questa fase creerai una vista denominata WLM_QUEUE_STATE_VW. Questa vista restituisce informazioni dalle tabelle di sistema seguenti.

- STV_WLM_CLASSIFICATION_CONFIG
- STV_WLM_SERVICE_CLASS_CONFIG
- STV_WLM_SERVICE_CLASS_STATE

Utilizzerai questa vista in tutto il tutorial per monitorare ciò che accade alle code dopo avere modificato la configurazione WLM. Nella seguente tabella sono descritti i dati restituiti dalla vista WLM_QUEUE_STATE_VW.

Colonna	Descrizione
coda	Il numero associato alla riga che rappresenta una coda. Il numero di coda determina l'ordine delle code nel database.
description	Valore che descrive se la coda è disponibile solo per determinati gruppi di utenti, determinati gruppi di query o per tutti i tipi di query.

Colonna	Descrizione
slots	Il numero di slot assegnate alla coda.
mem	La quantità di memoria, espressa in MB per slot, allocata alla coda.
max_execution_time	La quantità massima di tempo consentita per l'esecuzione di una query prima che venga terminata.
user_*	Valore che indica se è consentito l'utilizzo di caratteri jolly nella configura zione WLM per trovare la corrispondenza con i gruppi di utenti.
query_*	Valore che indica se è consentito l'utilizzo di caratteri jolly nella configura zione WLM per trovare la corrispondenza con i gruppi di query.
queued	Il numero di query che si trovano nella coda in attesa di essere elaborate.
executing	Il numero di query attualmente in esecuzione.
executed	Il numero di query che sono state eseguite.

Per creare la vista WLM_QUEUE_STATE_VW

1. Apri Amazon Redshift RSQL e connettiti al database di esempio TICKIT. Se questo database non è presente, vedi Prerequisiti.
2. Eseguire la query seguente per creare la vista WLM_QUEUE_STATE_VW.

```
create view WLM_QUEUE_STATE_VW as
select (config.service_class-5) as queue
, trim (class.condition) as description
, config.num_query_tasks as slots
, config.query_working_mem as mem
, config.max_execution_time as max_time
, config.user_group_wild_card as "user_*"
, config.query_group_wild_card as "query_*"
, state.num_queued_queries queued
, state.num_executing_queries executing
, state.num_executed_queries executed
from
STV_WLM_CLASSIFICATION_CONFIG class,
```

```
STV_WLM_SERVICE_CLASS_CONFIG config,
STV_WLM_SERVICE_CLASS_STATE state
where
class.action_service_class = config.service_class
and class.action_service_class = state.service_class
and config.service_class > 4
order by config.service_class;
```

3. Eseguire la query seguente per vedere le informazioni incluse nella vista.

```
select * from wlm_queue_state_vw;
```

Di seguito è riportato un risultato di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(querytype: any)	5	836	0	false	false	0	1	160

Fase 2: creazione della vista WLM_QUERY_STATE_VW

In questa fase creerai una vista denominata WLM_QUERY_STATE_VW. Questa vista restituisce informazioni dalla tabella di sistema STV_WLM_QUERY_STATE.

Utilizzerai questa vista in tutto il tutorial per monitorare le query in esecuzione. Nella seguente tabella sono descritti i dati restituiti dalla vista WLM_QUERY_STATE_VW.

Colonna	Descrizione
query	L'ID di query.
coda	Il numero della coda.
slot_count	Il numero di slot assegnate alla query.
start_time	L'ora in cui è stata avviata la query.
stato	Lo stato della query, ad esempio in esecuzione.
queue_time	Numero totale di microsecondi che la query ha trascorso nella coda.
exec_time	Numero di microsecondi durante i quali la query è stata in esecuzione.

Per creare la vista WLM_QUERY_STATE_VW

1. In RSQL esegui la query seguente per creare la vista WLM_QUERY_STATE_VW.

```
create view WLM_QUERY_STATE_VW as
select query, (service_class-5) as queue, slot_count, trim(wlm_start_time) as
 start_time, trim(state) as state, trim(queue_time) as queue_time, trim(exec_time) as
 exec_time
from stv_wlm_query_state;
```

2. Eseguire la query seguente per vedere le informazioni incluse nella vista.

```
select * from wlm_query_state_vw;
```

   Di seguito è riportato un risultato di esempio.

query	queue	slot_count	start_time	state	queue_time	exec_time
1249	1	1	2014-09-24 22:19:16	Executing	0	516

Fase 3: esecuzione delle query di test

In questa fase eseguirai query da più connessioni in RSQL ed esaminerai le tabelle di sistema per stabilire il modo in cui le query sono state instradate per l'elaborazione.

Per questo passaggio, è necessario che siano aperte due finestre RSQL:

- Nella finestra RSQL 1 eseguirai le query che monitorano lo stato delle code e le query che utilizzano le viste già create in questo tutorial.
- Nella finestra RSQL 2 eseguirai le query di lunga durata per modificare i risultati trovati nella finestra RSQL 1.

Per eseguire le query di test

1. Apri due finestre RSQL. Se è già aperta una finestra, sarà sufficiente aprire la seconda. È possibile utilizzare lo stesso account utente per entrambe le connessioni.
2. Nella finestra RSQL 1 esegui la query seguente.

```
select * from wlm_query_state_vw;
```

Di seguito è riportato un risultato di esempio.

query	queue	slot_count	start_time	state	queue_time	exec_time
1258	1	1	2014-09-24 22:21:03	Executing	0	549

Questa query restituisce un risultato autoreferenziale. La query attualmente in esecuzione è l'istruzione SELECT da questa vista. Una query su questa vista restituisce sempre almeno un risultato, che si deve confrontare con quello che si ottiene avviando la query di lunga durata al passaggio successivo.

3. Nella finestra RSQL 2 esegui una query dal database TICKIT di esempio. Questa query deve essere eseguita per circa un minuto, in modo che sia possibile esplorare i risultati della vista WLM_QUEUE_STATE_VW e la vista WLM_QUERY_STATE_VW creata precedentemente. In alcuni casi, potresti osservare che la query non viene eseguita abbastanza a lungo per interrogare entrambe le viste. In questi casi, puoi aumentare il valore del filtro su `l.listid` affinché la query venga eseguita più a lungo.

   > **Note**
   >
   > Per ridurre il tempo di esecuzione delle query e migliorare le prestazioni del sistema, Amazon Redshift memorizza i risultati di certi tipi di query nella memoria cache del nodo principale. Se è abilitato il caching dei risultati, l'esecuzione delle query successive è moto più rapida. Per evitare che la query venga eseguita troppo rapidamente, è possibile disabilitare il caching dei risultati per la sessione corrente.

   Per disattivare il caching dei risultati per la sessione corrente, è possibile impostare il parametro enable_result_cache_for_session su `off` come mostrato di seguito.

   ```
   set enable_result_cache_for_session to off;
   ```

   Nella finestra RSQL 2 esegui la query seguente.

   ```
   select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <
    100000;
   ```

4. Nella finestra RSQL 1, esegui una query WLM_QUEUE_STATE_VW e WLM_QUERY_STATE_VW e confronta i risultati con quelli precedenti.

```
select * from wlm_queue_state_vw;
select * from wlm_query_state_vw;
```

Di seguito sono riportati i risultati di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(querytype: any)	5	836	0	false	false	0	2	163

query	queue	slot_count	start_time	state	queue_time	exec_time
1267	1	1	2014-09-24 22:22:30	Executing	0	684
1265	1	1	2014-09-24 22:22:26	Executing	0	4080859

Di seguito vengono illustrate le differenze tra le query precedenti e i risultati di questa fase:

- Ora in WLM_QUERY_STATE_VW sono presenti due righe. Un risultato è la query autoreferenziale per l'esecuzione di un'operazione SELECT in questa vista. Il secondo risultato è la query di lunga durata della fase precedente.
- Il valore della colonna di esecuzione in WLM_QUEUE_STATE_VW è aumentato da 1 a 2. La voce di questa colonna indica che nella coda sono in esecuzione due query.
- Il valore della colonna eseguita aumenta ogni volta che si esegue una query nella coda.

La vista WLM_QUEUE_STATE_VW è utile per ottenere una vista generale delle code e per conoscere il numero di query in elaborazione in ogni coda. La vista WLM_QUERY_STATE_VW è utile per ottenere una vista più dettagliata delle singole query attualmente in esecuzione.

Sezione 2: modifica della configurazione delle code di query WLM

Ora che hai compreso il funzionamento delle code, puoi imparare come configurare le code di query utilizzando WLM manuale. In questa sezione creerai e configurerai un nuovo gruppo di parametri per il tuo cluster. Creerai due code utente aggiuntive e le configurerai in modo che accettino le query in base alle etichette del gruppo di utenti o del gruppo di query. Le query che non vengono instradate a una di queste due code verranno instradate alla coda predefinita in fase di runtime.

Per modificare la configurazione WLM in un gruppo di parametri

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.

2. Dal menu di navigazione scegliere Configurations (Configurazioni), quindi scegliere Workload management (Gestione carichi di lavoro) per visualizzare la pagina Workload management Gestione carichi di lavoro).
3. Scegli Create (Crea) per visualizzare la finestra Create parameter group (Crea gruppo di parametri).
4. Inserire **`WLMTutorial`** per Parameter group name (Nome gruppo di parametri) e Description (Descrizione), quindi scegliere Create (Crea) per creare il gruppo di parametri.

   > **Note**
   >
   > Il Parameter group name (Nome del gruppo di parametri) è trasformato in minuscolo al momento della creazione.

5. Nella pagina Workload management (Gestione workload), scegli il gruppo di parametri **`wlmtutorial`** per visualizzare la pagina dei dettagli con le schede Parameters (Parametri) e Workload management (Gestione workload).
6. Verificare di trovarti nella scheda Gestione del carico di lavoro, quindi scegliere Modifica modalità WLM per visualizzare la finestra Impostazioni simultaneità.
7. Scegli Manual WLM (WLM manuale), quindi scegli Save (Salva) per passare al WLM manuale.
8. Scegli Edit workload queues (Modifica code di workload).
9. Scegli Add queue (Aggiungi coda) due volte per aggiungere due code. Ora sono presenti tre code: Queue 1 (Coda 1), Queue 2 (Coda 2) e Default queue (Coda predefinita).
10. Inserisci le informazioni per ogni coda come segue:

    - Per Coda 1, inserire **`30`** alla voce Memoria (%), **`2`** alla voce Simultaneità su principale e **`test`** alla voce Gruppi di query. Lascia le altre impostazioni con i valori predefiniti.
    - Per Coda 2, inserire **`40`** alla voce Memoria (%), **`3`** alla voce Simultaneità su principalee **`admin`** alla voce Gruppi di utenti. Lascia le altre impostazioni con i valori predefiniti.
    - Non apportare alcuna modifica alla Default queue (Coda predefinita). WLM assegna memoria non allocata alla coda predefinita.

11. Per salvare le impostazioni, scegli Save (Salva).

Quindi, associa il gruppo di parametri che ha la configurazione WLM manuale a un cluster.

Associare un gruppo di parametri con configurazione WLM manuale a un cluster

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegliere Clusters (Cluster), quindi scegliere Clusters (Cluster) per visualizzare un elenco dei cluster.
3. Selezionare il cluster, come `examplecluster` per visualizzarne i dettagli. Quindi scegliere la scheda Proprietà per visualizzare le proprietà del cluster.
4. Nella sezione Configurazioni del database, scegliere Modifica, Modifica il gruppo di parametri per visualizzare la finestra dei gruppi di parametri.
5. Per Gruppi di parametri scegliere il gruppo di parametri **`wlmtutorial`** creato in precedenza.
6. Scegliere Salva le modifiche per associare il gruppo di parametri.

   Il cluster viene aggiornato con il gruppo di parametri modificato. Tuttavia, è necessario riavviare il cluster affinché le modifiche vengano applicate anche al database.
7. Scegliere il cluster, quindi selezionare Riavvia alla voce Operazioni.

Dopo il riavvio del cluster, il suo stato torna a essere Available (Disponibile).

Sezione 3: instradamento delle query alle code in base ai gruppi di utenti e ai gruppi di query

Ora il tuo cluster è associato a un nuovo gruppo di parametri e hai configurato WLM. Successivamente, eseguire alcune query per vedere come Amazon Redshift le instrada sulle code per l'elaborazione.

Fase 1: visualizzazione della configurazione delle code di query nel database

Verificare che la configurazione WLM del database sia quella prevista.

Per visualizzare la configurazione delle code di query

1. Apri RSQL ed esegui la query seguente. La query utilizza la vista WLM_QUEUE_STATE_VW creata in Fase 1: creazione della vista WLM_QUEUE_STATE_VW. Se prima del riavvio del cluster al database è già connessa una sessione, sarà necessario riconnetterla.

```
select * from wlm_queue_state_vw;
```

Di seguito è riportato un risultato di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(query group: test)	2	627	0	false	false	0	0	0
2	(user group: admin)	3	557	0	false	false	0	0	0
3	(querytype: any)	5	250	0	false	false	0	1	0

Confronta questi risultati on quelli ricevuti in Fase 1: creazione della vista WLM_QUEUE_STATE_VW. Come puoi notare, ora sono presenti due code aggiuntive. La coda 1 ora è la coda per il gruppo di query di verifica, mentre la coda 2 è quella per il gruppo di utenti amministratori.

La coda 3 è quella predefinita. L'ultima coda nell'elenco è sempre la coda predefinita. Si tratta della coda a cui vengono instradate le query se non è specificato un gruppo di utenti o di query.

2. Esegui la query seguente per verificare che venga eseguita nella coda 3.

```
select * from wlm_query_state_vw;
```

Di seguito è riportato un risultato di esempio.

query	queue	slot_count	start_time	state	queue_time	exec_time
2144	3	1	2014-09-24 23:49:59	Executing	0	550430

Fase 2: esecuzione di una query tramite la coda del gruppo di query

Per eseguire una query tramite la coda del gruppo di query

1. Eseguire la seguente query per instradarla al gruppo di query `test`.

```
set query_group to test;
select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <40000;
```

2. Dall'altra finestra RSQL esegui la query seguente.

```
select * from wlm_query_state_vw;
```

Di seguito è riportato un risultato di esempio.

query	queue	slot_count	start_time	state	queue_time	exec_time
2168	1	1	2014-09-24 23:54:18	Executing	0	6343309
2170	3	1	2014-09-24 23:54:24	Executing	0	847

La query è stata instradata al gruppo di query di verifica, che ora è la coda 1.

3. Seleziona tutto nella vista dello stato della coda.

```
select * from wlm_queue_state_vw;
```

Sarà visualizzato un risultato simile al seguente.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(query group: test)	2	627	0	false	false	0	1	0
2	(user group: admin)	3	557	0	false	false	0	0	0
3	(querytype: any)	5	250	0	false	false	0	1	3

4. Reimposta il gruppo di query ed esegui nuovamente la query lunga:

```
reset query_group;
select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <40000;
```

5. Eseguire le query sulle viste per vedere i risultati.

```
select * from wlm_queue_state_vw;
select * from wlm_query_state_vw;
```

Di seguito sono riportati i risultati di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(query group: test)	2	627	0	false	false	0	0	1
2	(user group: admin)	3	557	0	false	false	0	0	0
3	(querytype: any)	5	250	0	false	false	0	2	5

query	queue	slot_count	start_time	state	queue_time	exec_time
2186	3	1	2014-09-24 23:57:52	Executing	0	649
2184	3	1	2014-09-24 23:57:48	Executing	0	4137349

Il risultato dovrebbe indicare che la query è ora nuovamente in esecuzione nella coda 3.

Fase 3: creazione di un utente e un gruppo di database

Prima di poter eseguire una query in questa coda, dovrai creare il gruppo di utenti nel database e aggiungere un utente al gruppo. Quindi accederai con RSQL utilizzando le nuove credenziali utente ed eseguirai le query. Per creare utenti di database, è necessario eseguire le query come utente con privilegi avanzati, ad esempio l'amministratore.

Per creare un nuovo gruppo di utenti e un nuovo utente di database

1. Nel database crea un nuovo utente di database denominato `adminwlm` eseguendo il comando seguente in una finestra RSQL.

```
create user adminwlm createuser password '123Admin';
```

2. Quindi eseguire i comandi seguenti per creare il nuovo gruppo di utenti e aggiungervi il nuovo utente `adminwlm`.

```
create group admin;
alter group admin add user adminwlm;
```

Fase 4: esecuzione di una query tramite la coda del gruppo di utenti

Successivamente eseguirai una query e la instraderai alla coda del gruppo di utenti. Questa operazione viene eseguita quando si intende instradare la query a una coda configurata per gestire il tipo di query che si desidera eseguire.

Per eseguire una query tramite la coda del gruppo di utenti

1. Nella finestra RSQL 2 esegui le query seguenti per passare all'account `adminwlm` ed esegui una query con questo account.

```
set session authorization 'adminwlm';
select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <40000;
```

2. Nella finestra RSQL 1 esegui la query seguente per vedere la coda a cui vengono instradate le query.

```
select * from wlm_query_state_vw;
select * from wlm_queue_state_vw;
```

Di seguito sono riportati i risultati di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(query group: test)	2	627	0	false	false	0	0	1
2	(user group: admin)	3	557	0	false	false	0	1	0
3	(querytype: any)	5	250	0	false	false	0	1	8

query	queue	slot_count	start_time	state	queue_time	exec_time
2202	2	1	2014-09-25 00:01:38	Executing	0	4885796
2204	3	1	2014-09-25 00:01:43	Executing	0	650

La coda in viene eseguita questa query è la 2, ovvero la coda dell'utente `admin`. Le query che esegui avendo effettuato l'accesso come questo utente verranno seguite nella coda 2, a meno che non specifichi un gruppo di code diverso da utilizzare. La coda scelta dipende dalle regole di assegnazione delle code. Per ulteriori informazioni, consulta Regole di assegnazione delle code WLM.

3. Ora esegui la query seguente dalla finestra RSQL 2.

```
set query_group to test;
select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <40000;
```

4. Nella finestra RSQL 1 esegui la query seguente per vedere la coda a cui vengono instradate le query.

```
select * from wlm_queue_state_vw;
select * from wlm_query_state_vw;
```

Di seguito sono riportati i risultati di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(query group: test)	2	627	0	false	false	0	1	1
2	(user group: admin)	3	557	0	false	false	0	0	1
3	(querytype: any)	5	250	0	false	false	0	1	10

query	queue	slot_count	start_time	state	queue_time	exec_time
2218	1	1	2014-09-25 00:04:30	Executing	0	4819666
2220	3	1	2014-09-25 00:04:35	Executing	0	685

5. Al termine, reimposta il gruppo di query.

```
reset query_group;
```

Sezione 4: utilizzo di wlm_query_slot_count per ignorare temporaneamente il livello di simultaneità in una coda

Può accadere che gli utenti abbiano l'esigenza temporanea di un numero maggiore di risorse per una query specifica. In questo caso possono utilizzare l'impostazione di configurazione wlm_query_slot_count per ignorare temporaneamente il modo in cui gli slot vengono allocati in una coda di query. Gli slot sono unità di memoria e CPU utilizzati per elaborare le query. È possibile ignorare il numero di slot nel caso di query occasionali che richiedono grandi quantità di risorse nel cluster, ad esempio quando si esegue un'operazione VACUUM nel database.

Potresti trovare che gli utenti spesso devono impostare wlm_query_slot_count per determinati tipi di query. In tal caso, regola la configurazione WLM e fornisci agli utenti una coda più adatta alle esigenze delle loro query. Per ulteriori informazioni su come ignorare temporaneamente il livello di simultaneità utilizzando il numero di slot, consultare wlm_query_slot_count.

Fase 1: come ignorare il livello di simultaneità utilizzando wlm_query_slot_count

Ai fini del presente tutorial, eseguiremo la stessa query SELECT di lunga durata. La query verrà eseguita come utente `adminwlm` utilizzando wlm_query_slot_count per aumentare il numero di slot disponibili per la query.

Per ignorare il livello di simultaneità utilizzando wlm_query_slot_count

1. Aumentare il limite per la query per essere sicuri di disporre di tempo sufficiente per eseguire la query sulla vista WLM_QUERY_STATE_VW e vedere il risultato.

   ```
   set wlm_query_slot_count to 3;
   select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <40000;
   ```

2. Esegui la query WLM_QUERY_STATE_VW con l'utente amministratore per vederne l'esecuzione.

   ```
   select * from wlm_query_state_vw;
   ```

   Di seguito è riportato un risultato di esempio.

query	queue	slot_count	start_time	state	queue_time	exec_time
2240	2	3	2014-09-25 00:08:45	Executing	0	3731414
2242	3	1	2014-09-25 00:08:49	Executing	0	596

   Il numero di slot per la query è 3. Ciò significa che la query utilizza tutti e tre gli slot per l'elaborazione, allocando tutte le risorse nella coda alla query.

3. Ora esegui la seguente query.

```
select * from WLM_QUEUE_STATE_VW;
```

Di seguito è riportato un risultato di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(query group: test)	2	627	0	false	false	0	0	4
2	(user group: admin)	3	557	0	false	false	0	1	3
3	(querytype: any)	5	250	0	false	false	0	1	25

L'impostazione di configurazione di wlm_query_slot_count ha validità solo per la sessione corrente. Se la sessione scade o un altro utente esegue una query, viene utilizzata la configurazione WLM.

4. Reimpostare il numero di slot ed eseguire nuovamente il test.

```
reset wlm_query_slot_count;
select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <40000;
```

Di seguito sono riportati i risultati di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(query group: test)	2	627	0	false	false	0	0	2
2	(user group: admin)	3	557	0	false	false	0	1	2
3	(querytype: any)	5	250	0	false	false	0	1	14

query	queue	slot_count	start_time	state	queue_time	exec_time
2260	2	1	2014-09-25 00:12:11	Executing	0	4042618
2262	3	1	2014-09-25 00:12:15	Executing	0	680

Fase 2: esecuzione delle query da sessioni diverse

A questo punto eseguire le query da sessioni diverse.

Per eseguire le query da sessioni diverse

1. Nelle finestre RSQL 1 e 2 esegui la query seguente per utilizzare il gruppo di query di test.

```
set query_group to test;
```

2. Nella finestra RSQL 1 esegui la seguente query di lunga durata.

```
select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <40000;
```

3. Poiché la query di lunga durata è ancora in esecuzione nella finestra RSQL 1, esegui quanto segue. Questi comandi aumentano il numero di slot per utilizzare tutti gli slot per la coda, quindi avviano l'esecuzione della query di lunga durata.

```
set wlm_query_slot_count to 2;
select avg(l.priceperticket*s.qtysold) from listing l, sales s where l.listid <40000;
```

4. Apri una terza finestra RSQL ed esegui una query sulle viste per vedere i risultati.

```
select * from wlm_queue_state_vw;
select * from wlm_query_state_vw;
```

Di seguito sono riportati i risultati di esempio.

queue	description	slots	mem	max_time	user_*	query_*	queued	executing	executed
0	(super user) and (query group: superuser)	1	357	0	false	false	0	0	0
1	(query group: test)	2	627	0	false	false	1	1	2
2	(user group: admin)	3	557	0	false	false	0	0	3
3	(querytype: any)	5	250	0	false	false	0	1	18

query	queue	slot_count	start_time	state	queue_time	exec_time
2286	1	2	2014-09-25 00:16:48	QueuedWaiting	3758950	0
2282	1	1	2014-09-25 00:16:33	Executing	0	19335850
2288	3	1	2014-09-25 00:16:52	Executing	0	666

Nota che la prima query utilizza uno degli slot assegnati alla coda 1 per eseguire la query. Inoltre, tieni presente che una query è in attesa nella coda (dove `queued` è `1` e `state` è `QueuedWaiting`). Una volta completata la prima query, inizierà l'esecuzione della seconda. Ciò accade perché entrambe le query sono instradate al gruppo di query `test` e la seconda deve attendere che siano disponibili slot sufficienti per iniziare l'elaborazione.

Sezione 5: pulizia delle risorse

Il tuo cluster genera dei costi fino a che è in esecuzione. Una volta completato questo tutorial, ripristinare lo stato precedente dell'ambiente seguendo la procedura descritta in Scoprire risorse aggiuntive e reimpostare l'ambiente nella Guida alle operazioni di base di Amazon Redshift.

Per ulteriori informazioni sulla funzionalità WLM, consultare Implementazione della gestione del carico di lavoro.

Utilizzo del dimensionamento della simultaneità

Con la funzione dimensionamento simultaneo, puoi supportare migliaia di utenti e di query simultanee, con prestazioni di query a velocità costante. Quando si attiva il dimensionamento simultaneo, Amazon Redshift aggiunge automaticamente ulteriore capacità del cluster quando necessario per elaborare un aumento delle query di lettura e di scrittura. Gli utenti visualizzano sempre i dati più recenti, indipendentemente dal fatto che le query vengano eseguite nel cluster principale o in un cluster a dimensionamento simultaneo.

Puoi gestire le query inviate al cluster di dimensionamento simultaneo configurando le code WLM. Quando si attiva il dimensionamento simultaneo per una coda, anziché attendere in una coda le query idonee vengono inviate al cluster di dimensionamento simultaneo.

I cluster a dimensionamento simultaneo vengono addebitati solo per il tempo in cui eseguono le query. Per ulteriori informazioni sui prezzi, tra cui il modo in cui i costi si accumulano e i costi minimi, consulta Prezzi di Concurrency Scaling.

Capacità di dimensionamento simultaneo

Quando si attiva il dimensionamento simultaneo per una coda WLM, questo funziona per operazioni di lettura, ad esempio le query del pannello di controllo. Funziona anche per le operazioni di scrittura comunemente utilizzate, come le istruzioni per l'importazione e l'elaborazione dei dati.

Funzionalità di dimensionamento simultaneo per le operazioni di scrittura

Il dimensionamento simultaneo supporta le operazioni di scrittura più utilizzate, come istruzioni di estrazione, trasformazione e caricamento (ETL). Il dimensionamento simultaneo per le operazioni di scrittura è particolarmente utile quando si desidera mantenere tempi di risposta coerenti quando il cluster riceve un numero elevato di richieste. Migliora la velocità effettiva per le operazioni di scrittura contendendo le risorse nel cluster principale.

Il dimensionamento simultaneo supporta le istruzioni COPY, INSERT, DELETE, UPDATE e CREATE TABLE AS (CTAS). Inoltre, il dimensionamento simultaneo supporta l'aggiornamento delle viste materializzate che non utilizzano aggregazioni. Altre istruzioni DML (Data Manipulation Language) e DDL (Data Definition Language) non sono supportate. Quando le istruzioni di scrittura non supportate, come CREATE senza TABLE AS, vengono incluse in una transazione esplicita prima delle istruzioni di scrittura supportate, nessuna delle istruzioni di scrittura verrà eseguita sui cluster con dimensionamento simultaneo.

Quando si accumula credito per il dimensionamento simultaneo, questo accumulo si applica sia alle operazioni di lettura che a quelle di scrittura.

Limitazioni per il dimensionamento simultaneo

Di seguito sono riportate le limitazioni per l'utilizzo del dimensionamento simultaneo di Amazon Redshift:

- Non supporta query su tabelle che utilizzano chiavi di ordinamento interlacciato.
- Non supporta query sulle tabelle temporanee.
- Non supporta query che accedono a risorse esterne protette da reti restrittive o configurazioni Virtual Private Cloud (VPC).
- Non supporta query che contengono funzioni Python e funzioni Lambda definite dall'utente (UDF).
- Non supporta query che accedono alle tabelle di sistema, alle tabelle di catalogo PostgreSQL o alle tabelle di no-backup.
- Non supporta le query COPY o UNLOAD che accedono a una risorsa esterna quando sono disponibili autorizzazioni restrittive per le policy IAM. Ciò include le autorizzazioni applicate alla risorsa, come un bucket Amazon S3 o una tabella DynamoDB, o all'origine. Le fonti IAM possono includere quanto segue:
  - `aws:sourceVpc`— Un VPC sorgente.
  - `aws:sourceVpce`— Un endpoint VPC di origine.
  - `aws:sourceIp`— Un indirizzo IP di origine.

  In alcuni casi, potrebbe essere necessario rimuovere le autorizzazioni che limitano la risorsa o l'origine, in modo che le query COPY e UNLOAD che accedono alla risorsa vengano inviate al cluster di scalabilità simultanea.

  Per ulteriori informazioni sulle politiche delle risorse, consulta Tipi di policy nella guida per l' AWS Identity and Access Management utente e Controllo dell'accesso dagli endpoint VPC con le policy bucket.
- Il dimensionamento simultaneo di Amazon Redshift per le operazioni di scrittura non è supportato per le operazioni DDL, ad esempio CREATE TABLE o ALTER TABLE.
- Non supporta ANALYZE per il comando COPY.
- Non supporta le operazioni di scrittura su una tabella di destinazione in cui DISTSTYLE è impostato su ALL.

- Non supporta COPY dai seguenti formati di file:
  - Parquet
  - ORC
- Non supporta le operazioni di scrittura su tabelle con colonne di identità.
- Amazon Redshift supporta il dimensionamento simultaneo per le operazioni di scrittura solo su nodi RA3 di Amazon Redshift, in particolare ra3.16xlarge, ra3.4xlarge e ra3.xlplus. Il dimensionamento simultaneo per le operazioni di scrittura non è supportato su altri tipi di nodi.

Regioni AWS per la scalabilità simultanea

La scalabilità simultanea è disponibile nelle seguenti regioni: AWS

- Regione Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Regione Stati Uniti orientali (Ohio) (us-east-2)
- AWS GovCloud (Stati Uniti orientali)
- Regione Stati Uniti occidentali (California settentrionale) (us-west-1)
- Regione Stati Uniti occidentali (Oregon) (us-west-2)
- Regione Asia Pacifico (Mumbai) (ap-south-1)
- Regione Asia Pacifico (Seoul) (ap-northeast-2)
- Regione Asia Pacifico (Singapore) (ap-southeast-1)
- Regione Asia Pacifico (Sydney) (ap-southeast-2)
- Regione Asia Pacifico (Tokyo) (ap-northeast-1)
- Regione Canada (Centrale) (ca-central-1)
- Regione Europa (Francoforte) (eu-central-1)
- Regione Europa (Irlanda) (eu-west-1)
- Regione Europa (Londra) (eu-west-2)
- Regione Europa (Parigi) (eu-west-3)
- Regione Europa (Stoccolma) (eu-north-1)
- Regione Sud America (San Paolo) (sa-east-1)

Candidati per il dimensionamento simultaneo

Le query vengono instradate al cluster a dimensionamento simultaneo solo quando il cluster principale soddisfa i seguenti requisiti:

- Piattaforma EC2-VPC.
- Il tipo di nodo deve essere dc2.8xlarge, dc2.large, ra3.xlplus, ra3.4xlarge o ra3.16xlarge. Il dimensionamento simultaneo per le operazioni di scrittura è supportato solo su nodi RA3 di Amazon Redshift, in particolare ra3.16xlarge, ra3.4xlarge e ra3.xlplus.
- Massimo 32 nodi di calcolo per cluster con tipi di nodi ra3.xlplus, ra3.4xlarge o ra3.16xlarge. Inoltre, il numero di nodi del cluster principale non può essere maggiore di 32 al momento della creazione del cluster originale. Ad esempio, anche se un cluster ha attualmente 20 nodi, ma è stato originariamente creato con 40, non soddisfa i requisiti per il dimensionamento simultaneo. Al contrario, se un cluster DC2 ha attualmente 40 nodi, ma è stato originariamente creato con 20, soddisfa i requisiti per la scalabilità simultanea.
- Non un cluster a nodo singolo.

Configurazione delle code di dimensionamento simultaneo

Le query vengono instradate ai cluster di dimensionamento simultaneo abilitando una coda di gestione del carico di lavoro come coda di dimensionamento simultaneo. Per abilitare il dimensionamento simultaneo su una coda, impostare il valore Modalità di dimensionamento simultaneo su auto.

Quando il numero di query instradate a una coda di dimensionamento simultaneo supera la concorrenza configurata della coda, le query idonee vengono inviate al cluster di dimensionamento simultaneo. Quando gli slot diventano disponibili, le query vengono eseguite nel cluster principale. Il numero di code è limitato solo dal numero di code consentite per cluster. Come con qualsiasi coda di gestione del carico di lavoro, le query vengono instradate a una coda di dimensionamento simultaneo in base ai gruppi di utenti o all'etichettatura di query con etichette di gruppi di query. Puoi anche instradare le query definendo Regole di monitoraggio delle query WLM. Ad esempio, potresti instradare tutte le query che impiegano più di 5 secondi a una coda di dimensionamento simultaneo.

Il numero predefinito di cluster di dimensionamento simultaneo è uno. Il numero di cluster di dimensionamento simultaneo che è possibile utilizzare è controllato da max_concurrency_scaling_clusters.

Monitoraggio del dimensionamento simultaneo

È possibile visualizzare se una query è in esecuzione nel cluster principale o in un cluster a dimensionamento simultaneo selezionando Cluster dalla console Amazon Redshift e scegliendo un cluster. Quindi scegli la scheda Monitoraggio delle query e Simultaneità del carico di lavoro per visualizzare le informazioni sulle query in esecuzione e sulle query in coda.

Per trovare i tempi di esecuzione, eseguire una query sulla tabella STL_QUERY e filtrare la colonna `concurrency_scaling_status`. La seguente query confronta il tempo di attesa in coda e il tempo di esecuzione per le query eseguite nel cluster a dimensionamento simultaneo e le query eseguite nel cluster principale.

```
SELECT w.service_class AS queue
, CASE WHEN q.concurrency_scaling_status = 1 THEN 'concurrency scaling cluster' ELSE
 'main cluster' END as concurrency_scaling_status
, COUNT( * ) AS queries
, SUM( q.aborted ) AS aborted
, SUM( ROUND( total_queue_time::NUMERIC / 1000000,2) ) AS queue_secs
, SUM( ROUND( total_exec_time::NUMERIC / 1000000,2) ) AS exec_secs
FROM stl_query q
JOIN stl_wlm_query w
USING (userid,query)
WHERE q.userid > 1
AND q.starttime > '2019-01-04 16:38:00'
AND q.endtime < '2019-01-04 17:40:00'
GROUP BY 1,2
ORDER BY 1,2;
```

Regola i valori `starttime` e `endtime` in base alle tue esigenze.

Visualizzazioni di sistema per il dimensionamento simultaneo

Una serie di visualizzazioni di sistema con il prefisso SVCS fornisce i dettagli dalle tabelle di log di sistema relativi alle query nei cluster principale e a dimensionamento simultaneo.

Le seguenti viste contengono informazioni simili alle viste STL o SVL corrispondenti:

- SVCS_ALERT_EVENT_LOG
- SVCS_COMPILE
- SVCS_EXPLAIN

- SVCS_PLAN_INFO
- SVCS_QUERY_SUMMARY
- SVCS_STREAM_SEGS

Le seguenti visualizzazioni sono specifiche del dimensionamento simultaneo.

- SVCS_CONCURRENCY_SCALING_USAGE

Per ulteriori informazioni sul dimensionamento simultaneo, consulta i seguenti argomenti nella Guida alla gestione di Amazon Redshift.

- Visualizzazione dei dati di dimensionamento della concorrenza
- Visualizzazione delle prestazioni del cluster durante l'esecuzione delle query
- Visualizzazione dei dettagli delle query

Utilizzo dall'accelerazione di query brevi

L'accelerazione di query brevi (SQA) rende prioritarie le query a esecuzione breve rispetto a quelle a esecuzione prolungata. L'accelerazione di query brevi (SQA, Short Query Acceleration) esegue query a esecuzione breve in uno spazio dedicato, in modo che le query SQA non siano costrette ad attendere in coda dietro query più lunghe. SQA assegna la priorità solo alle query che hanno un'esecuzione breve e si trovano in una coda definita dall'utente. Con SQA, le query a esecuzione breve iniziano l'esecuzione più rapidamente e gli utenti visualizzano i risultati prima.

Se abiliti SQA, è possibile ridurre le code di gestione del carico di lavoro (WLM) dedicate all'esecuzione di query brevi. Inoltre, le query a esecuzione prolungata non devono contendere gli slot con le query brevi in una coda, quindi puoi configurare le code WLM per utilizzare un numero inferiore di slot di query. Quando usi una simultaneità inferiore, il throughput delle query aumenta e le prestazioni generali del sistema risultano migliorate per la maggior parte dei carichi di lavoro.

Le istruzioni CREATE TABLE AS (CTAS) e le query di sola lettura, ad esempio le istruzioni SELECT, sono idonee per SQA.

Amazon Redshift utilizza un algoritmo di machine learning per analizzare ciascuna query idonea e prevedere il tempo di esecuzione della query. Per impostazione predefinita, WLM assegna dinamicamente un valore per il tempo di esecuzione massimo SQA in base all'analisi del carico

di lavoro del cluster. In alternativa, è possibile specificare un valore fisso di 1-20 secondi. Se il tempo di esecuzione previsto della query è inferiore al runtime massimo SQA definito o assegnato dinamicamente e la query è in attesa in una coda WLM, SQA separa la query dalle code WLM e la pianifica per l'esecuzione prioritaria. Se una query viene eseguita più a lungo rispetto al tempo di esecuzione massimo SQA, WLM sposta la query nella prima coda WLM corrispondente in base alle regole di assegnazione delle code WLM. Nel corso del tempo le previsioni migliorano man mano che SQA assimila i tuoi modelli di query.

SQA è abilitato per impostazione predefinita nel gruppo di parametri predefiniti e per tutti i nuovi gruppi di parametri. Per disabilitare SQA nella console Amazon Redshift, modificare la configurazione WLM per un gruppo di parametri e deselezionare Abilita accelerazione di query brevi. Come best practice, ti consigliamo di utilizzare un numero di slot di query WLM pari o inferiore a 15 per mantenere le prestazioni generali del sistema a livello ottimale. Per informazioni sulla modifica delle configurazioni WLM, consulta Configurazione della gestione del carico di lavoro nella Guida alla gestione di Amazon Redshift.

Tempo di esecuzione massimo per query brevi

Quando abiliti SQA, WLM imposta il tempo di esecuzione massimo per le query brevi su dinamico per impostazione predefinita. Ti consigliamo di mantenere le impostazioni dinamiche per il tempo massimo di esecuzione SQA. È possibile ignorare l'impostazione di default specificando un valore fisso compreso tra 1 e 20 secondi.

In alcuni casi, puoi prendere in considerazione l'utilizzo di valori diversi per i valori massimi del tempo di esecuzione SQA per migliorare le prestazioni del sistema. In questi casi, analizza il tuo carico di lavoro per trovare il tempo massimo di esecuzione per la maggior parte delle tue query a esecuzione breve. La seguente query restituisce il tempo di esecuzione massimo per le query a circa il 70° percentile.

```
select least(greatest(percentile_cont(0.7)
within group (order by total_exec_time / 1000000) + 2, 2), 20)
from stl_wlm_query
where userid >= 100
and final_state = 'Completed';
```

Dopo aver identificato un valore di tempo di esecuzione massimo adatto al carico di lavoro, non è necessario modificarlo a meno che il carico di lavoro non cambi significativamente.

Monitoraggio dell'accelerazione di query brevi (SQA, Short Query Acceleration)

Per controllare se SQA è abilitato, esegui la seguente query. Se la query restituisce una riga, SQA è abilitato.

```
select * from stv_wlm_service_class_config
where service_class = 14;
```

La seguente query mostra il numero di query che hanno sostato in ciascuna coda di query (classe di servizio). Mostra anche il tempo medio di esecuzione, il numero di query con tempo di attesa al 90° percentile e il tempo medio di attesa. Le query SQA vengono utilizzate nella classe di servizio 14.

```
select final_state, service_class, count(*), avg(total_exec_time),
percentile_cont(0.9) within group (order by total_queue_time), avg(total_queue_time)
from stl_wlm_query where userid >= 100 group by 1,2 order by 2,1;
```

Per trovare quali query sono state rilevate da SQA e completate, esegui la seguente query.

```
select a.queue_start_time, a.total_exec_time, label, trim(querytxt)
from stl_wlm_query a, stl_query b
where a.query = b.query and a.service_class = 14 and a.final_state = 'Completed'
order by b.query desc limit 5;
```

Per trovare le query che SQA ha rilevato ma che hanno generato un timeout, esegui la seguente query.

```
select a.queue_start_time, a.total_exec_time, label, trim(querytxt)
from stl_wlm_query a, stl_query b
where a.query = b.query and a.service_class = 14 and a.final_state = 'Evicted'
order by b.query desc limit 5;
```

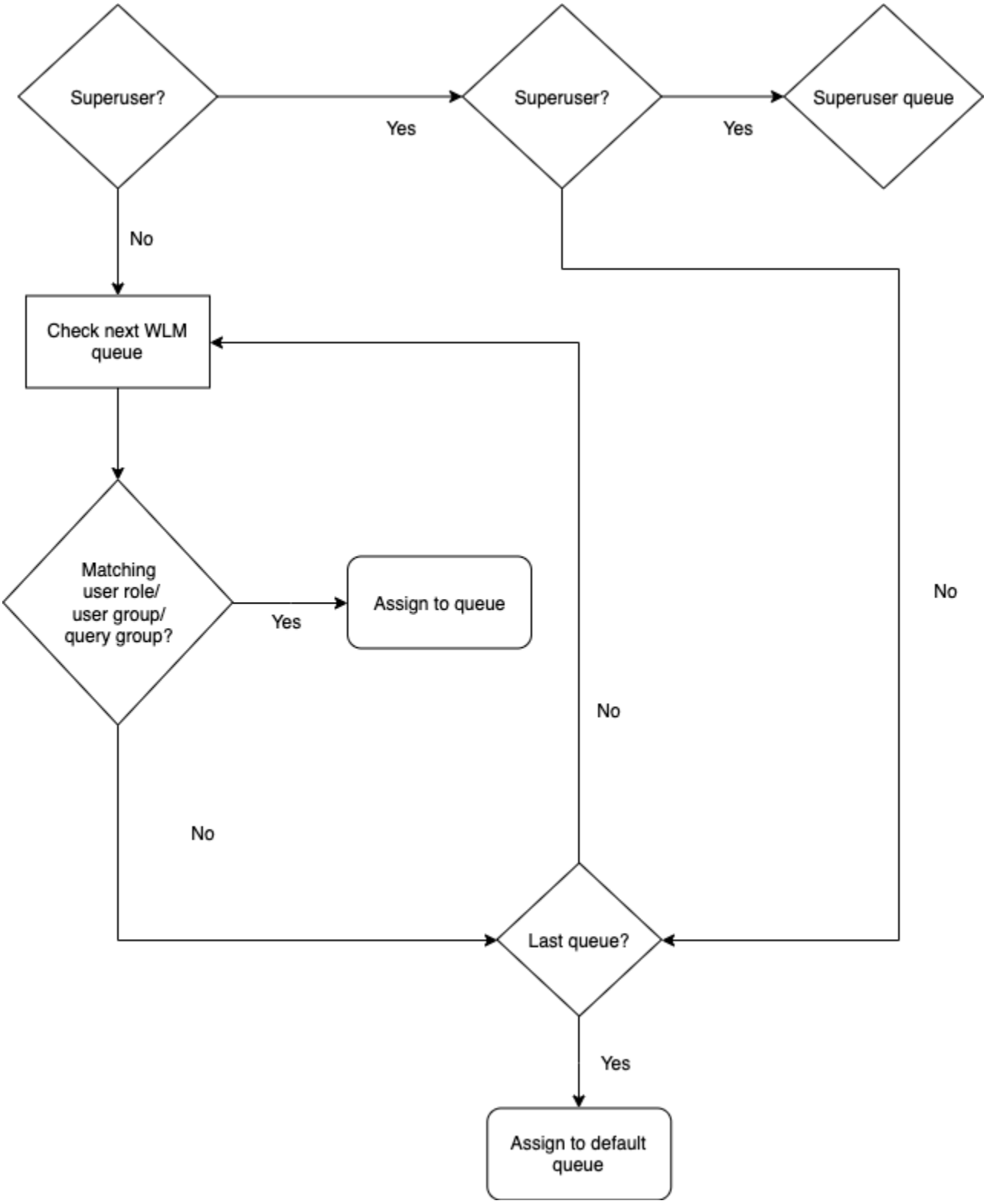
Per ulteriori informazioni sulle query rifiutate e, più in generale, sulle operazioni basate su regole che possono essere eseguite sulle query, consulta Regole di monitoraggio delle query WLM.

Regole di assegnazione delle code WLM

Quando un utente esegue una query, WLM assegna la query alla prima coda corrispondente, in base alle regole di assegnazione delle code WLM:

1. Se un utente ha effettuato l'accesso come utente con privilegi avanzati ed esegue una query nel gruppo di query con l'etichetta superuser, la query viene assegnata alla coda dell'utente con privilegi avanzati.
2. Se un utente fa parte di un ruolo, appartiene a un gruppo di utenti elencato o esegue una query in un gruppo di utenti elencato, la query viene assegnata alla prima coda corrispondente.
3. Se una query non soddisfa alcun criterio, viene assegnata alla coda predefinita, ossia l'ultima definita nella configurazione WLM.

Lo schema seguente illustra il funzionamento di queste regole.

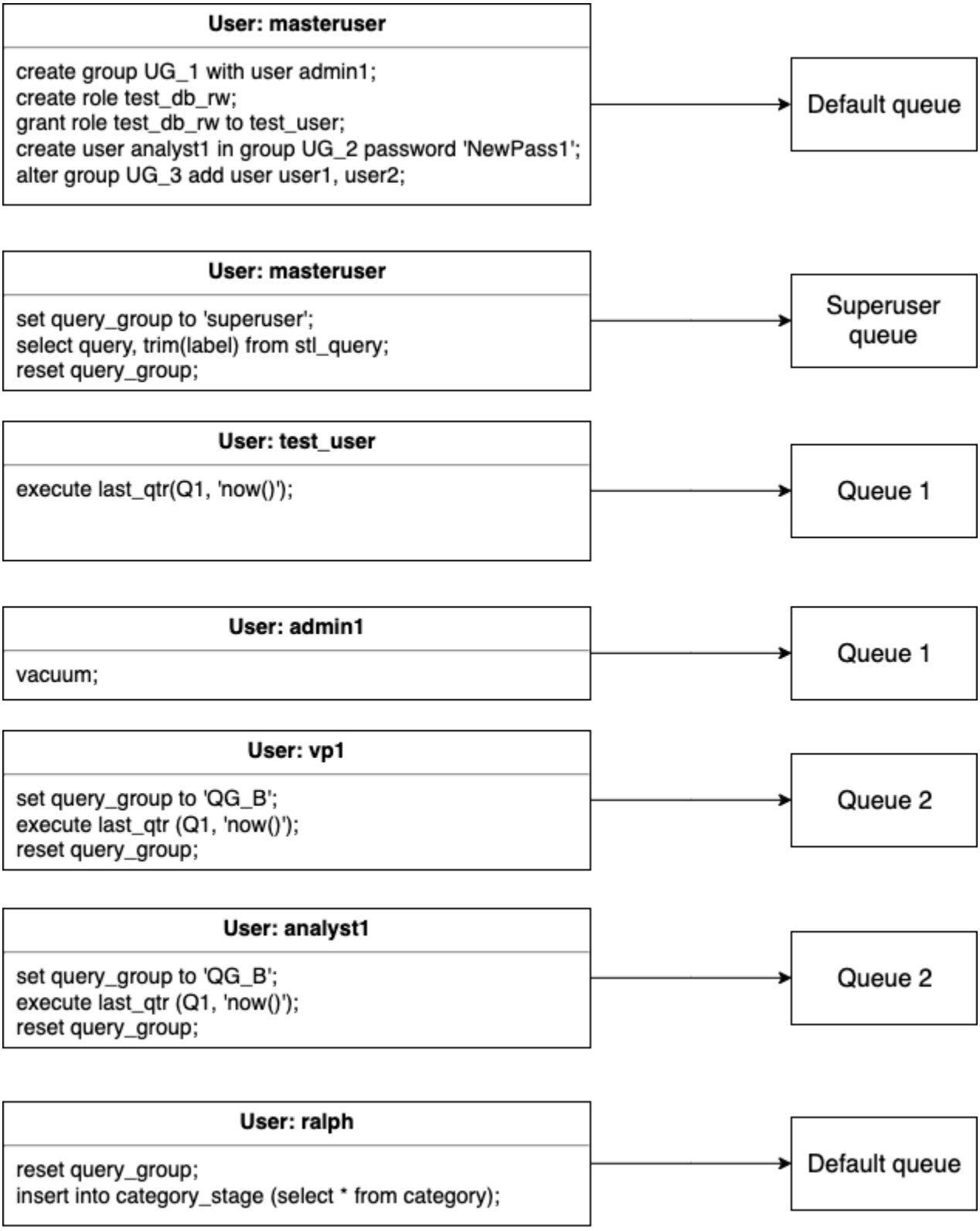

Superuser?
Yes
Superuser?
Yes
Superuser queue
No
Check next WLM queue
Matching user role/ user group/ query group?
Yes
Assign to queue
No
No
No
Last queue?
Yes
Assign to default queue

Esempio di assegnazione delle code

La seguente tabella mostra una configurazione WLM con la coda dell'utente con privilegi avanzati e quattro code definite dall'utente.

Queue	Simultaneità	Ruoli utente	Gruppi di utenti	Gruppi di query
Superuser	1			superuser
1	5	test_db_rw	UG_1	
2	5			QG_B
3	5		UG_2	QG_C
Predefinita	5			

L'immagine seguente mostra come le query vengono assegnate alle code della tabella precedente in base ai gruppi di utenti e ai gruppi di query. Per informazioni su come assegnare le query ai gruppi di utenti e ai gruppi di query in fase di runtime, vedi Assegnazione delle query alle code più avanti in questa sezione.



In questo esempio WLM esegue le seguenti assegnazioni:

1. La prima serie di istruzioni mostra tre modi per assegnare gli utenti ai gruppi di utenti. Le istruzioni sono eseguite dall'utente `adminuser` che non è un membro di un gruppo di utenti elencato in una coda WLM. Nessun gruppo di query è impostato, quindi le istruzioni vengono indirizzate alla coda predefinita.
2. L'utente `adminuser` è un utente con privilegi avanzati e il gruppo di query è impostato su `'superuser'`, pertanto la query viene assegnata alla coda dell'utente con privilegi avanzati.
3. All'utente `test_user` è assegnato il ruolo `test_db_rw` elencato nella coda 1, pertanto la query viene assegnata alla coda 1.
4. L'utente `admin1` è un membro del gruppo di utenti elencato nella coda 1, pertanto la query viene assegnata alla coda 1.
5. L'utente `vp1` non è un membro di alcun gruppo di utenti elencato. Il gruppo di query è impostato su `'QG_B'`, quindi la query viene assegnata alla coda 2.
6. L'utente `analyst1` è un membro del gruppo di utenti elencato nella coda 3, ma `'QG_B'` corrisponde alla coda 2, quindi la query viene assegnata alla coda 2.
7. L'utente `ralph` non è un membro di alcun gruppo di utenti elencato e il gruppo di query è stato reimpostato, pertanto non è presente alcuna coda corrispondente. La query viene assegnata alla coda predefinita.

Assegnazione delle query alle code

I seguenti esempi consentono di assegnare le query alle code in base ai ruoli degli utenti, ai gruppi di utenti e ai gruppi di query.

Assegnazione delle query alle code in base ai ruoli degli utenti

Se a un utente è assegnato un ruolo e tale ruolo è collegato a una coda, le query eseguite da quell'utente vengono assegnate a tale coda. L'esempio seguente crea un ruolo utente denominato `sales_rw` e assegna l'utente `test_user` a quel ruolo.

```
create role sales_rw;
grant role sales_rw to test_user;
```

Puoi anche combinare le autorizzazioni di due ruoli assegnando esplicitamente un ruolo a un altro ruolo. L'assegnazione di un ruolo nidificato a un utente concede all'utente le autorizzazioni di entrambi i ruoli.

```
create role sales_rw;
create role sales_ro;
grant role sales_ro to role sales_rw;
grant role sales_rw to test_user;
```

Per visualizzare l'elenco degli utenti a cui sono stati concessi ruoli nel cluster, esegui una query sulla tabella SVV_USER_GRANTS. Per visualizzare l'elenco dei ruoli a cui sono stati concessi ruoli nel cluster, esegui una query sulla tabella SVV_ROLE_GRANTS.

```
select * from svv_user_grants;
select * from svv_role_grants;
```

Assegnazione delle query alle code in base ai gruppi di utenti

Se il nome di un gruppo di utenti è elencato in una definizione di coda, le query eseguite dai membri di quel gruppo di utenti vengono assegnate alla coda corrispondente. L'esempio seguente crea gruppi di utenti e aggiungere utenti ai gruppi utilizzando i comandi SQL CREA UTENTE, CREATE GROUP e ALTER GROUP.

```
create group admin_group with user admin246, admin135, sec555;
create user vp1234 in group ad_hoc_group password 'vpPass1234';
alter group admin_group add user analyst44, analyst45, analyst46;
```

Assegnazione di una query a un gruppo di utenti

Puoi assegnare una query a una coda in fase di runtime assegnando la query al gruppo di query appropriato. Utilizza il comando SET per iniziare un gruppo di query.

```
SET query_group TO group_label
```

Qui, *group_label* è un'etichetta del gruppo di query elencata nella configurazione WLM.

Tutte le query eseguite dopo il comando `SET query_group` vengono eseguite come membri del gruppo di query specificato fino a quando non reimposti il gruppo di query o termini la sessione di accesso corrente. Per informazioni sull'impostazione e sulla reimpostazione di oggetti Amazon Redshift, consultare SET e RESET nella documentazione di riferimento relativa ai comandi SQL.

Le etichette del gruppo di query specificate devono essere incluse nella configurazione WLM corrente; in caso contrario, il comando SET query_group non ha alcun effetto sulle code di query.

L'etichetta definita nella clausola TO viene acquisita nei log delle query in modo da poterla utilizzare per la risoluzione dei problemi. Per informazioni sul parametro di configurazione query_group, vedi query_group nella documentazione di riferimento relativa alla configurazione.

L'esempio seguente esegue due query come parte del gruppo di query 'priority' e quindi reimposta il gruppo di query.

```
set query_group to 'priority';
select count(*)from stv_blocklist;
select query, elapsed, substring from svl_qlog order by query desc limit 5;
reset query_group;
```

Assegnazione delle query alla coda dell'utente con privilegi avanzati

Per assegnare una query alla coda dell'utente con privilegi avanzati, accedere ad Amazon Redshift come utente con privilegi avanzati ed eseguire la query nel gruppo di utenti con privilegi avanzati. Al termine, reimposta il gruppo di query in modo che le query successive non vengano eseguite nella coda dell'utente con privilegi avanzati.

L'esempio seguente assegna due comandi da eseguire nella coda dell'utente con privilegi avanzati.

```
set query_group to 'superuser';

analyze;
vacuum;
reset query_group;
```

Per visualizzare un elenco di utenti con privilegi avanzati, eseguire una query sulla tabella del catalogo di sistema PG_USER.

```
select * from pg_user where usesuper = 'true';
```

Proprietà di configurazione dinamiche e statiche WLM

Le proprietà della configurazione WLM sono dinamiche o statiche. Puoi applicare proprietà dinamiche al database senza riavviare il cluster, ma le proprietà statiche richiedono un riavvio del cluster affinché le modifiche abbiano effetto. Tuttavia, se si modificano contemporaneamente le proprietà dinamiche e statiche, è necessario riavviare il cluster affinché tutte le modifiche diventino effettive (indipendentemente dal tipo di proprietà).

Durante l'applicazione delle proprietà dinamiche, lo stato dei cluster è `modifying`. Il passaggio tra WLM automatico e WLM manuale è una modifica statica che per essere effettiva richiede l'esecuzione di un riavvio del cluster.

La tabella seguente indica quali proprietà WLM sono dinamiche o statiche quando si utilizza WLM automatico o WLM manuale.

Proprietà WLM	WLM automatico	WLM manuale
Gruppi di query	Dinamico	Statico
Carattere jolly per gruppi di query	Dinamico	Statico
Gruppi di utenti	Dinamico	Statico
Carattere jolly per gruppi di utenti	Dinamico	Statico
Ruoli utente	Dinamico	Statico
Carattere jolly del ruolo utente	Dinamico	Statico
Simultaneità nel cluster principale	Non applicabile	Dinamico
Modalità Dimensionamento simultaneo	Dinamico	Dinamico
Abilitazione dell'accelerazione di query brevi	Non applicabile	Dinamico
Tempo di esecuzione massimo per query brevi	Dinamico	Dinamico
Percentuale di memoria da utilizzare	Non applicabile	Dinamico
Timeout	Non applicabile	Dinamico

Proprietà WLM	WLM automatico	WLM manuale
Priority (Priorità)	Dinamico	Non applicabile
Aggiunta o rimozione di code	Dinamico	Statico

Se si modifica una regola di monitoraggio delle query (QMR), la modifica viene eseguita automaticamente senza la necessità di modificare il cluster.

> **Note**
>
> Quando utilizzi WLM manuale, se il valore di timeout viene modificato, il nuovo valore è applicato a tutte le query la cui esecuzione inizia dopo la modifica del valore. Se la simultaneità o la percentuale di memoria da utilizzare viene modificata, Amazon Redshift passa alla nuova configurazione in modo dinamico. Di conseguenza, le query attualmente in esecuzione non sono coinvolte dalla modifica. Per ulteriori informazioni, consultare Allocazione della memoria dinamica WLM.

Argomenti

Allocazione della memoria dinamica WLM

In ogni coda, WLM crea un numero di slot di query pari al livello di simultaneità della coda. La quantità di memoria allocata a uno slot di query equivale alla percentuale di memoria allocata alla coda divisa per il conteggio degli slot. Se si modifica la simultaneità o l'allocazione della memoria, Amazon Redshift gestisce in modo dinamico la transizione alla nuova configurazione WLM. Pertanto, le query attive possono essere eseguite fino al completamento utilizzando la quantità di memoria correntemente allocata. Allo stesso tempo, Amazon Redshift si accerta che l'utilizzo totale della memoria non superi mai il 100% della memoria disponibile.

Il gestore del carico di lavoro utilizza il seguente processo per gestire la transizione:

1. WLM ricalcola l'allocazione della memoria per ogni nuovo slot di query.

2. Se uno slot di query non viene utilizzato attivamente da una query in esecuzione, WLM rimuove lo slot, rendendo disponibile la memoria per i nuovi slot.
3. Se uno slot di query è utilizzato attivamente, WLM attende la fine della query.
4. Quando le query attive vengono completate, gli slot vuoti vengono rimossi e la memoria associata viene liberata.
5. Man mano che la memoria sufficiente diventa disponibile per aggiungere uno o più slot, i nuovi slot vengono aggiunti.
6. Al termine di tutte le query che erano in esecuzione al momento della modifica, il numero degli slot equivale al nuovo livello di simultaneità e la transizione alla nuova configurazione WLM è completa.

In effetti, le query in esecuzione quando la modifica ha effetto continuano a utilizzare l'allocazione originale della memoria. Le query che sono in coda quando la modifica ha effetto vengono indirizzate ai nuovi slot non appena diventano disponibili.

Se le proprietà dinamiche WLM vengono modificate durante il processo di transizione, WLM inizia immediatamente la transizione alla nuova configurazione, a partire dallo stato corrente. Per visualizzare lo stato della transizione, esegui la query sulla tabella di sistema STV_WLM_SERVICE_CLASS_CONFIG.

Esempio di WLM dinamico

Supponiamo che il WLM del cluster sia configurato con due code tramite le seguenti proprietà dinamiche.

Queue	Simultaneità	% di memoria da usare
1	4	50%
2	4	50%

Ora supponiamo che il tuo cluster abbia 200 GB di memoria disponibile per l'elaborazione delle query. Questo numero è arbitrario e viene utilizzato solo a scopo illustrativo. Come mostra la seguente equazione, a ogni slot sono allocati 25 GB.

```
(200 GB * 50% ) / 4 slots  = 25 GB
```

Successivamente, modifichi WLM per utilizzare le seguenti proprietà dinamiche.

Queue	Simultaneità	% di memoria da usare
1	3	75%
2	4	25%

Come mostra la seguente equazione, la nuova allocazione di memoria per ogni slot nella coda 1 è di 50 GB.

```
(200 GB * 75% ) / 3 slots = 50 GB
```

Supponiamo che le query A1, A2, A3 e A4 siano in esecuzione quando viene applicata la nuova configurazione e le query B1, B2, B3 e B4 siano in coda. WLM riconfigura dinamicamente gli slot delle query come segue.

Fase	Esecuzione di query	Conteggio slot corrente	Conteggio slot target	Memoria allocata	Memoria disponibile
1	A1, A2, A3, A4	4	0	100 GB	50 GB
2	A2, A3, A4	3	0	75 GB	75 GB
3	A3, A4	2	0	50 GB	100 GB
4	A3, A4, B1	2	1	100 GB	50 GB
5	A4, B1	1	1	75 GB	75 GB
6	A4, B1, B2	1	2	125 GB	25 GB
7	B1, B2	0	2	100 GB	50 GB
8	B1, B2, B3	0	3	150 GB	0 GB

1. WLM ricalcola l'allocazione della memoria per ogni slot di query. Originariamente, alla coda 1 erano allocati 100 GB. La nuova coda ha una allocazione totale di 150 GB e pertanto ha immediatamente 50 GB disponibili. La coda 1 ora utilizza quattro slot e il nuovo livello di simultaneità è tre slot, quindi non vengono aggiunti nuovi slot.
2. Al termine di una query, lo slot viene rimosso e vengono liberati 25 GB. La coda 1 ora ha tre slot e 75 GB di memoria disponibile. La nuova configurazione richiede 50 GB per ogni nuovo slot, ma il nuovo livello di simultaneità è tre slot, quindi non vengono aggiunti nuovi slot.
3. Al termine di una seconda query, lo slot viene rimosso e vengono liberati 25 GB. La coda 1 ora ha due slot e 100 GB di memoria disponibile.
4. Viene aggiunto un nuovo slot utilizzando 50 GB di memoria disponibile. La coda 1 ora ha tre slot e 50 GB di memoria disponibile. Le query in coda possono ora essere indirizzate al nuovo slot.
5. Al termine di una terza query, lo slot viene rimosso e vengono liberati 25 GB. La coda 1 ora ha due slot e 75 GB di memoria disponibile.
6. Viene aggiunto un nuovo slot utilizzando 50 GB di memoria disponibile. La coda 1 ora ha tre slot e 25 GB di memoria disponibile. Le query in coda possono ora essere indirizzate al nuovo slot.
7. Al termine di una quarta query, lo slot viene rimosso e vengono liberati 25 GB. La coda 1 ora ha due slot e 50 GB di memoria disponibile.
8. Viene aggiunto un nuovo slot utilizzando 50 GB di memoria disponibile. La coda 1 ora ha tre slot con 50 GB ciascuno e tutta la memoria disponibile è stata allocata.

La transizione è completa e tutti gli slot di query sono disponibili per le query in coda.

Regole di monitoraggio delle query WLM

Nella gestione del carico di lavoro (WLM) di Amazon Redshift, le regole di monitoraggio delle query definiscono i limiti delle prestazioni basati sui parametri per le code WLM e specificano l'operazione da eseguire quando una query oltrepassa tali limiti. Ad esempio, per una coda dedicata alle query di breve durata, puoi creare una regola che annulla le query eseguite per più di 60 secondi. Per tracciare le query mal progettate, potresti avere un'altra regola che registra le query che contengono cicli annidati.

Le regole di monitoraggio delle query vengono definite come parte della configurazione della gestione del carico di lavoro (WLM). Puoi definire fino a 25 regole per ogni coda, con un limite di 25 regole per tutte le code. Ogni regola include fino a tre condizioni o predicati e un'azione. Un predicato è formato da un parametro, una condizione di confronto (=, < o >) e un valore. Se tutti i predicati di una

regola sono soddisfatti, viene attivata l'azione di quella regola. Le possibili azioni delle regole sono registrazione, hop e interruzione, come discusso di seguito.

Le regole in una determinata coda si applicano solo alle query in esecuzione in quella coda. Una regola è indipendente dalle altre regole.

WLM calcola i parametri ogni 10 secondi. Se durante lo stesso periodo vengono attivate più regole, WLM avvia l'operazione più grave: interrompere, quindi eseguire l'hop, quindi registrare. Se l'azione è hop o interruzione, viene registrata e la query viene rimossa dalla coda. Se l'azione è registrazione, la query continua l'esecuzione nella coda. WLM avvia una sola azione di registrazione per query per regola. Se la coda contiene altre regole, tali regole rimangono in vigore. Se l'azione è hop e la query viene instradata su un'altra coda, si applicano le regole della nuova coda. Per ulteriori informazioni sul monitoraggio delle query e sulle operazioni di tracciamento effettuate su query specifiche, consulta la raccolta di esempi all'indirizzo Utilizzo dall'accelerazione di query brevi.

Quando tutti i predicati di una regola vengono soddisfatti, WLM scrive una riga nella tabella di sistema STL_WLM_RULE_ACTION. Inoltre, Amazon Redshift registra i parametri delle query per le query attualmente in esecuzione su STV_QUERY_METRICS. I parametri delle query completate sono archiviati in STL_QUERY_METRICS.

Definizione di una regola di monitoraggio delle query

Le regole di monitoraggio delle query vengono create come parte della configurazione WLM definita nella definizione del gruppo di parametri del cluster.

È possibile creare regole utilizzando AWS Management Console o utilizzando JSON a livello di codice.

> **Note**
>
> Se scegli di creare regole a livello di programmazione, ti consigliamo vivamente di utilizzare la console per generare il JSON da includere nella definizione del gruppo di parametri. Per ulteriori informazioni, consulta Creazione o modifica di una regola di monitoraggio delle query con la console e Configurazione dei valori dei parametri tramite la AWS CLI nella Guida alla gestione di Amazon Redshift.

Per definire una regola di monitoraggio della query, è necessario specificare i seguenti elementi:

- Un nome di regola: i nomi delle regole devono essere univoci all'interno della configurazione WLM. Possono essere composti da un massimo di 32 caratteri alfanumerici o caratteri di sottolineatura e non possono contenere spazi o virgolette. Puoi avere fino a 25 regole per coda, che è anche il limite complessivo di tutte le code.
- Uno o più predicati: ogni regola può avere fino a tre predicati. Se tutti i predicati di una regola sono soddisfatti, viene attivata l'azione associata. Un predicato è definito da un nome parametro, un operatore (=, < o >) e un valore. Un esempio è `query_cpu_time > 100000`. Per un elenco di parametri ed esempi di valori per parametri diversi, vedi Metriche per il monitoraggio delle query per Amazon Redshift dopo questa sezione.
- Un'operazione: se sono attivate più regole, WLM sceglie la regola con l'operazione più severa. Le possibili azioni, in ordine crescente di gravità, sono:
  - Registra: registra le informazioni sulla query nella tabella di sistema STL_WLM_RULE_ACTION. Usa l'azione di registrazione quando vuoi scrivere solo un record di log. WLM crea al massimo una registrazione per query per regola. A seguito di un'azione di registrazione, restano in vigore altre regole e WLM continua a monitorare la query.
  - Hop (disponibile solo con WLM manuale): registra l'azione e passa la query alla coda corrispondente successiva. Se non esiste alcuna coda corrispondente, la query viene annullata. QMR salta solo le istruzioni CREATE TABLE AS (CTAS) e le query di sola lettura, ad esempio le istruzioni SELECT. Per ulteriori informazioni, consultare Hop della coda di query WLM.
  - Interrompi: registra l'azione e annulla la query. Le istruzioni COPY e le operazioni di manutenzione, come ANALYZE e VACUUM, non vengono interrotte da QMR.
  - Modifica priorità (disponibile solo con WLM automatica): modifica la priorità di una query.

Per limitare l'esecuzione di query, ti consigliamo di creare una regola di monitoraggio di query anziché utilizzare un timeout WLM. Ad esempio, puoi impostare `max_execution_time` su 50.000 millisecondi, come illustrato nel seguente frammento di codice JSON.

```
"max_execution_time": 50000
```

Ti consigliamo piuttosto di definire una regola di monitoraggio di query equivalente che imposta `query_execution_time` a 50 secondi come indicato nel seguente frammento di codice JSON:

```
"rules":
[
    {
        "rule_name": "rule_query_execution",
```

```
        "predicate": [
            {
                "metric_name": "query_execution_time",
                "operator": ">",
                "value": 50
            }
        ],
        "action": "abort"
    }
]
```

Per le fasi per creare o modificare una regola di monitoraggio delle query, consulta Creazione o modifica di una regola di monitoraggio delle query con la console e Proprietà nel parametro wlm_json_configuration nella Guida alla gestione di Amazon Redshift.

Ulteriori informazioni sulle regole di monitoraggio delle query sono disponibili nei seguenti argomenti:

- Metriche per il monitoraggio delle query per Amazon Redshift
- Modelli di regole di monitoraggio delle query
- Creazione di una regola con la console
- Configurazione della gestione del carico di lavoro
- Tabelle e viste di sistema per le regole di monitoraggio delle query

Metriche per il monitoraggio delle query per Amazon Redshift

Nella tabella seguente vengono descritti i parametri usati nelle regole di monitoraggio delle query. Questi parametri sono diversi da quelli memorizzati nelle tabelle di sistema STV_QUERY_METRICS e STL_QUERY_METRICS.

Per un determinato parametro, la soglia delle prestazioni viene tracciata a livello di query o di segmento. Per ulteriori informazioni sui segmenti e sulle fasi, consultare Pianificazione di query e flusso di lavoro di esecuzione.

> **Note**
>
> Il parametro Timeout WLM è diverso dalle regole di monitoraggio delle query.

Parametro	Nome	Descrizione
Tempo CPU query	`query_cpu_time`	Tempo di CPU utilizzato dalla query, in secondi. `CPU time` è diverso da `Query execution time`. I valori validi sono compresi nell'intervallo 0-999.999.
Blocchi letti	`query_blocks_read`	Numero di blocchi di dati da 1 MB letti dalla query. I valori validi sono compresi nell'intervallo 0-1.048.575.
Analizza conteggio righe	`scan_row_count`	Il numero di righe in una fase di scansione. Il conteggio delle righe è il numero totale di righe generate prima di aver applicato filtri alle righe contrassegnate per l'eliminazione (righe fantasma) e prima dell'applicazione di filtri di query definiti dall'utente. I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.
Tempo di esecuzione query	`query_execution_time`	Tempo di esecuzione trascorso per una query, in secondi. Il tempo di esecuzione non include il tempo trascorso in attesa in una coda. I valori validi sono compresi nell'intervallo 0-86.399.
Tempo nella coda di query	`query_queue_time`	Tempo trascorso in attesa in coda, in secondi. I valori validi sono compresi nell'intervallo 0-86.399.
Utilizzo CPU	`query_cpu_usage_percent`	Percentuale di capacità della CPU utilizzata dalla query.

Parametro	Nome	Descrizione
		I valori validi sono compresi nell'intervallo 0-6.399.
Memoria su disco	`query_temp_blocks_to_disk`	Spazio su disco temporaneo utilizzato per scrivere risultati intermedi, in blocchi da 1 MB. I valori validi sono compresi nell'intervallo 0-319.815.679.
Differenza CPU	`cpu_skew`	Il rapporto tra l'utilizzo di CPU massimo per ogni sezione e l'utilizzo di CPU medio per tutte le sezioni. Questo parametro viene definito a livello di segmento. I valori validi sono compresi nell'intervallo 0-99.
Differenza I/O	`io_skew`	Il rapporto tra i blocchi massimi letti (I/O) per ogni sezione di blocchi medi letti per tutte le sezioni. Questo parametro viene definito a livello di segmento. I valori validi sono compresi nell'intervallo 0-99.
Righe unite	`join_row_count`	Il numero di righe elaborate in una fase di unione. I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.
Conteggio righe unione loop nidificati	`nested_loop_join_row_count`	Il numero o le righe di un'unione loop nidificate. I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.
Restituisci conteggio righe	`return_row_count`	Il numero di righe restituite dalla query. I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.

Parametro	Nome	Descrizione
Tempo di esecuzione segmento	`segment_execution_time`	Tempo di esecuzione trascorso per un singolo segmento, in secondi. Per evitare o ridurre gli errori di campionamento, includi `segment_execution_time > 10` nelle regole. I valori validi sono compresi nell'intervallo 0-86.388.
Analizza conteggio righe Spectrum	`spectrum_scan_row_count`	Il numero di righe di dati in Amazon S3 scansionate da una query di Amazon Redshift Spectrum. I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.
Dimensione analisi Spectrum	`spectrum_scan_size_mb`	La dimensione in MB dei dati in Amazon S3 scansionati da una query Amazon Redshift Spectrum. I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.
Priorità delle query	`query_priority`	La priorità della query. I valori validi sono `HIGHEST`, `HIGH`, `NORMAL`, `LOW` e `LOWEST`. Quando confronti `query_priority` utilizzando gli operator maggiore di (**>**) e inferiore a (**<**) operators, `HIGHEST` è maggiore di `HIGH`, `HIGH` è maggiore di `NORMAL` e così via.

Note

- L'operazione hop non è supportata con il predicato `query_queue_time`. Vale a dire che le regole definite per l'hop quando un predicato `query_queue_time` viene soddisfatto vengono ignorate.

- I tempi di esecuzione dei segmenti brevi possono causare errori di campionamento con alcuni parametri, ad esempio `io_skew` e `query_cpu_usage_percent`. Per evitare o ridurre gli errori di campionamento, includi il tempo di esecuzione dei segmenti nelle regole. Un buon punto di partenza è `segment_execution_time > 10`.

La vista SVL_QUERY_METRICS mostra i parametri per le query completate. La vista SVL_QUERY_METRICS_SUMMARY mostra i valori massimi dei parametri per le query completate. Utilizza i valori in queste viste come aiuto per determinare i valori di soglia per la definizione delle regole di monitoraggio delle query.

Metriche per il monitoraggio delle query per Amazon Redshift serverless

Nella tabella seguente vengono descritti i parametri utilizzati nelle regole per il monitoraggio delle query per Amazon Redshift serverless.

Parametro	Nome	Descrizione
Tempo CPU query	`max_query_cpu_time`	Tempo di CPU utilizzato dalla query, in secondi. `CPU time` è diverso da `Query execution time`. I valori validi sono compresi nell'intervallo 0-999.999.
Blocchi letti	`max_query_blocks_read`	Numero di blocchi di dati da 1 MB letti dalla query. I valori validi sono compresi nell'intervallo 0-1.048.575.
Analizza conteggio righe	`max_scan_row_count`	Il numero di righe in una fase di scansione. Il conteggio delle righe è il numero totale di righe generate prima di aver applicato filtri alle righe contrassegnate per l'eliminazione (righe fantasma) e prima dell'applicazione di filtri di query definiti dall'utente.

Parametro	Nome	Descrizione
		I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.
Tempo di esecuzione query	max_query_execution_time	Tempo di esecuzione trascorso per una query, in secondi. Il tempo di esecuzione non include il tempo trascorso in attesa in una coda. Se una query supera il tempo di esecuzione impostato, Amazon Redshift Serverless la arresta. I valori validi sono compresi nell'intervallo 0-86.399.
Tempo nella coda di query	`max_query_queue_time`	Tempo trascorso in attesa in coda, in secondi. I valori validi sono compresi nell'intervallo 0-86.399.
Utilizzo CPU	`max_query_cpu_usage_percent`	Percentuale di capacità della CPU utilizzata dalla query. I valori validi sono compresi nell'intervallo 0-6.399.
Memoria su disco	`max_query_temp_blocks_to_disk`	Spazio su disco temporaneo utilizzato per scrivere risultati intermedi, in blocchi da 1 MB. I valori validi sono compresi nell'intervallo 0-319.815.679.
Righe unite	`max_join_row_count`	Il numero di righe elaborate in una fase di unione. I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.

Parametro	Nome	Descrizione
Conteggio righe unione loop nidificati	`max_nested_loop_join_row_count`	Il numero o le righe di un'unione loop nidificate. I valori validi sono compresi nell'intervallo 0-999.999.999.999.999.

> **Note**
>
> - L'operazione hop non è supportata con il predicato `max_query_queue_time`. Vale a dire che le regole definite per l'hop quando un predicato `max_query_queue_time` viene soddisfatto vengono ignorate.
> - I tempi di esecuzione dei segmenti brevi possono causare errori di campionamento con alcuni parametri, ad esempio `max_io_skew` e `max_query_cpu_usage_percent`.

Modelli di regole di monitoraggio delle query

Quando si aggiunge una regola tramite la console Amazon Redshift, è possibile scegliere di creare una regola da un modello predefinito. Amazon Redshift crea una nuova regola con un set di predicati e popola i predicati con i valori di default. L'azione predefinita è registrazione. Puoi modificare i predicati e l'azione per soddisfare il tuo caso d'uso.

La tabella seguente elenca i modelli disponibili.

Nome modello	Predicati	Descrizione
Nested loop join	`nested_loop_join_row_count > 100`	Un'unione loop nidificata potrebbe indicare un predicato di unione incompleto che spesso determina la restituzione di un set molto grande (un prodotto cartesiano). Utilizza un conteggio di riga basso per trovare in anticipo una query con potenzialmente un eccessivo tempo di esecuzione.

Nome modello	Predicati	Descrizione
La query restituisce un numero elevato di righe	`return_row_count > 1000000`	Se dedichi una coda a query semplici a esecuzione breve, è possibile includere una regola che trova le query che restituiscono un conteggio elevato di righe. Il modello utilizza un valore predefinito di 1 milione di righe. Per alcuni sistemi, potresti considerare un milione di righe un conteggio elevato oppure in un sistema più grande, un miliardo o più di righe potrebbero essere un conteggio elevato.
Esegui l'unione con un numero elevato di righe	`join_row_count > 1000000000`	Una fase di unione che comporta un numero insolitamente elevato di righe potrebbe indicare la necessità di filtri più restrittivi. Il modello utilizza un valore predefinito di 1 miliardo di righe. Per una coda ad hoc (una tantum) destinata a query semplici e veloci, è possibile utilizzare un numero inferiore.
Utilizzo del disco intensivo durante la scrittura di risultati intermedi	`query_temp_blocks_to_disk > 100000`	Quando le query attualmente in esecuzione e utilizzano più RAM di sistema di quella disponibile, il motore di esecuzione della query scrive i risultati intermedi sul disco (memoria vuota). In genere, questa condizione è il risultato di una query di tipo rogue che di solito è anche la query che utilizza la maggior parte dello spazio su disco. La soglia accettabile per l'utilizzo del disco varia in base al tipo di nodo del cluster e al numero di nodi. Il modello utilizza un valore predefinito di 100.000 blocchi o 100 GB. Per un piccolo cluster, è possibile utilizzare un numero inferiore.

Nome modello	Predicati	Descrizione
Query di lunga durata con differenza I/O elevata	`segment_execution_time > 120` e `io_skew > 1.30`	La differenza I/O si verifica quando una sezione del nodo ha una velocità I/O molto più alta rispetto alle altre sezioni. Come punto di partenza, una differenza di 1,30 (1,3 volte la media) è considerata alta. La differenza I/O elevata non è sempre un problema, ma se combinata con un tempo di query a esecuzione prolungata potrebbe indicare un problema relativo allo stile di distribuzione o alla chiave di ordinamento.

Tabelle e viste di sistema per le regole di monitoraggio delle query

Quando tutti i predicati di una regola vengono soddisfatti, WLM scrive una riga nella tabella di sistema STL_WLM_RULE_ACTION. Questa riga contiene i dettagli della query che ha attivato la regola e l'operazione derivante.

Inoltre, Amazon Redshift registra i parametri per le query per le seguenti tabelle e viste di sistema.

- La tabella STV_QUERY_METRICS visualizza i parametri per le query attualmente in esecuzione.
- La tabella STL_QUERY_METRICS registra i parametri per le query completate.
- La vista SVL_QUERY_METRICS mostra i parametri per le query completate.
- La vista SVL_QUERY_METRICS_SUMMARY mostra i valori massimi dei parametri per le query completate.

Tabelle e viste di sistema di WLM

WLM configura le code delle query secondo le classi di servizio WLM, definite internamente. Amazon Redshift crea diverse code interne in base a queste classi di servizio insieme alle code definite nella configurazione WLM. I termini coda e classe di servizio sono spesso utilizzati in modo intercambiabile nelle tabelle di sistema. La coda dell'utente con privilegi avanzati utilizza la classe di servizio 5. Le code definite dall'utente utilizzano la classe di servizio 6 e successive.

Puoi visualizzare lo stato delle query, le code e le classi di servizio utilizzando le tabelle di sistema specifiche di WLM. Esegui le query sulle seguenti tabelle di sistema per effettuare quanto segue:

- Visualizzare quali query vengono tracciate e quali risorse sono allocate dal gestore del carico di lavoro.
- Vedere a quale coda è stata assegnata una query.
- Visualizzare lo stato di una query che viene attualmente monitorata dal gestore del carico di lavoro.

Nome tabella	Descrizione
STL_WLM_ERROR	Contiene un log degli eventi di errore relativi a WLM.
STL_WLM_QUERY	Elenca le query tracciate da WLM.
STV_WLM_CLASSIFICATION_CONFIG	Mostra le regole di classificazione correnti per WLM.
STV_WLM_QUERY_QUEUE_STATE	Registra lo stato corrente delle code di query.
STV_WLM_QUERY_STATE	Fornisce uno snapshot dello stato corrente delle query che vengono monitorate da WLM.
STV_WLM_QUERY_TASK_STATE	Contiene lo stato corrente delle attività di query.
STV_WLM_SERVICE_CLASS_CONFIG	Registra le configurazioni delle classi di servizio per WLM.
STV_WLM_SERVICE_CLASS_STATE	Contiene lo stato corrente delle classi di servizio.
STL_WLM_RULE_ACTION	Registra i dettagli sulle operazioni derivanti dalle regole di monitoraggio di query WLM associate alle code definite dall'utente.
STV_WLM_QMR_CONFIG	Registra la configurazione per le regole di monitoraggio di query WLM (QMR).

Utilizzi l'ID attività per tenere traccia di una query nelle tabelle di sistema. L'esempio seguente illustra come ottenere l'ID attività della query utente inviata più di recente:

```
select task from stl_wlm_query where exec_start_time =(select max(exec_start_time) from
 stl_wlm_query);

task
------
137
(1 row)
```

L'esempio seguente visualizza le query attualmente in esecuzione o in attesa in varie classi di servizio (code). Questa query è utile per tracciare il carico di lavoro complessivo per Amazon Redshift:

```
select * from stv_wlm_query_state order by query;


xid |task|query|service_| wlm_start_  |  state  |queue_ | exec_
    |    |     |class   | time        |         |time   | time
----+----+-----+--------+-------------+---------+-------+--------
2645| 84 | 98  | 3      | 2010-10-... |Returning|   0   | 3438369
2650| 85 | 100 | 3      | 2010-10-... |Waiting  |   0   | 1645879
2660| 87 | 101 | 2      | 2010-10-... |Executing|   0   | 916046
2661| 88 | 102 | 1      | 2010-10-... |Executing|   0   | 13291
(4 rows)
```

ID classe di servizio WLM

La tabella seguente elena gli ID assegnati alle classi di servizio.

ID	Classe di servizio
1-4	Riservate per il sistema.
5	Utilizzata dalla coda dell'utente con privilegi avanzati.
6-13	Utilizzate dalle code WLM manuale definite nella configura zione WLM.

ID	Classe di servizio
14	Utilizzata dall'accelerazione di query brevi.
15	Riservata alle operazioni di manutenzione eseguite da Amazon Redshift.
100-107	Utilizzata dalla coda WLM automatico se auto_wlm è true.

Gestione della sicurezza del database

Argomenti

Puoi gestire la sicurezza del database controllando gli utenti che possono accedere agli oggetti del database e gli oggetti nello specifico.

L'accesso agli oggetti di database dipende dalle autorizzazioni fornite agli utenti o ai gruppi. Nelle linee guida seguenti viene riepilogato il funzionamento della sicurezza del database:

- Di default, le autorizzazioni vengono concesse solo al proprietario dell'oggetto.
- Gli utenti del database di Amazon Redshift sono utenti denominati che possono connettersi a un database. Le autorizzazioni vengono fornite a un utente in due modi: in modo esplicito, ovvero mediante assegnazione diretta all'account, oppure in modo implicito, in quanto membro di un gruppo a cui sono concesse le autorizzazioni.
- I gruppi sono raccolte di utenti a cui possono essere assegnate le autorizzazioni collettivamente per una più agevole gestione della sicurezza.
- Gli schemi sono raccolte di tabelle di database e altri oggetti di database. Gli schemi sono simili alle directory del file system, con la differenza che non possono essere nidificati. Agli utenti si può concedere l'accesso a uno o più schemi.

Inoltre, Amazon Redshift utilizza le seguenti funzionalità per fornire un controllo più preciso su quali utenti hanno accesso a quali oggetti del database:

- Il controllo degli accessi basato sui ruoli (RBAC) consente di assegnare autorizzazioni ai ruoli che è possibile poi applicare agli utenti, consentendo di controllare le autorizzazioni per grandi gruppi di utenti. A differenza dei gruppi, i ruoli possono ereditare le autorizzazioni da altri ruoli.

  La sicurezza a livello di riga (RLS) consente di definire policy che limitano l'accesso alle righe di propria scelta, quindi applicarle a utenti o gruppi.

  Il mascheramento dinamico dei dati (DDM) protegge ulteriormente i dati trasformandoli durante il runtime della query in modo da consentire agli utenti di accedere ai dati senza esporre dettagli sensibili.

Per esempi di implementazione della sicurezza, vedi Esempio di controllo dell'accesso di utenti e gruppi.

Per ulteriori informazioni sulla protezione dei dati, consulta Sicurezza in Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Panoramica della sicurezza di Amazon Redshift

La sicurezza del database Amazon Redshift è diversa da altri tipi di sicurezza di Amazon Redshift. Oltre alla sicurezza del database descritta in questa sezione, Amazon Redshift offre le caratteristiche seguenti per gestire la sicurezza:

- Credenziali di accesso: l'accesso alla console di gestione Amazon AWS Redshift è controllato dalle AWS autorizzazioni del tuo account. Per ulteriori informazioni, consultare Credenziali di accesso.
- Gestione degli accessi: per controllare l'accesso a risorse Amazon Redshift specifiche, definisci account AWS Identity and Access Management (IAM). Per ulteriori informazioni, consultare Controllo dell'accesso alle risorse di Amazon Redshift.
- Gruppi di sicurezza del cluster: per concedere ad altri utenti l'accesso in entrata a un cluster Amazon Redshift, è necessario definire un gruppo di sicurezza del cluster e associarlo a un cluster. Per maggiori informazioni, consultare Gruppi di sicurezza dei cluster Amazon Redshift.
- VPC: per proteggere l'accesso al cluster utilizzando un ambiente di rete virtuale, è possibile avviare il cluster in un Amazon Virtual Private Cloud (VPC). Per ulteriori informazioni, consultare Gestione dei cluster in Virtual Private Cloud (VPC).

- Crittografia cluster: per crittografare i dati in tutte le tabelle create dall'utente, è possibile abilitare la crittografia del cluster quando si avvia il cluster. Per ulteriori informazioni, consultare Cluster Amazon Redshift.
- Connessioni SSL: per crittografare la connessione tra il client SQL e il cluster, è possibile utilizzare la crittografia Secure Sockets Layer (SSL). Per ulteriori informazioni, consultare Connessione al cluster mediante SSL.
- Crittografia dei dati di caricamento: per crittografare i file di dati del caricamento della tabella quando vengono caricati su Amazon S3, è possibile usare la crittografia lato server o la crittografia lato client. Quando si esegue il caricamento dai dati crittografati lato server, Amazon S3 gestisce la decrittografia in modo trasparente. Quando si esegue il caricamento dai dati crittografati lato client, il comando COPY di Amazon Redshift decritta i dati quando carica la tabella. Per ulteriori informazioni, consulta Caricamento di dati crittografati su Amazon S3.
- Dati in transito: per proteggere i dati in transito nel AWS cloud, Amazon Redshift utilizza SSL con accelerazione hardware per comunicare con Amazon S3 o Amazon DynamoDB per operazioni di COPY, UNLOAD, backup e ripristino.
- Controllo degli accessi a livello di colonna: per avere il controllo degli accessi a livello di colonna per i dati in Amazon Redshift, utilizzare le istruzioni di concessione e revoca a livello di colonna senza dover implementare il controllo degli accessi basato sulle viste o utilizzare un altro sistema.
- Controllo di sicurezza a livello di riga: per avere un controllo di sicurezza a livello di riga per i dati in Amazon Redshift, crea e collega a ruoli o utenti policy che limitano l'accesso alle righe definite.

Autorizzazioni utente di default per il database

Quando crei un oggetto database, ne diventi proprietario. Di default, solo un utente con privilegi avanzati o il proprietario di un oggetto può modificarlo, eseguire query o concedere autorizzazioni per l'oggetto. Per consentire agli utenti di utilizzare un oggetto, è necessario concedere le autorizzazioni necessarie all'utente o al gruppo in cui è incluso. Gli utenti con privilegi avanzati del database dispongono delle stesse autorizzazioni dei proprietari del database.

Amazon Redshift supporta le seguenti autorizzazioni: SELECT, INSERT, UPDATE, DELETE, REFERENCES, CREATE, TEMPORARY e USAGE. Autorizzazioni diverse sono associate a tipi di oggetto diversi. Per informazioni sulle autorizzazioni degli oggetti di database supportate da Amazon Redshift, consulta il comando GRANT.

Solo il proprietario ha il permesso di modificare o distruggere un oggetto.

Di default, tutti gli utenti hanno le autorizzazioni CREATE e USAGE per lo schema PUBLIC di un database. Per impedire agli utenti di creare oggetti nello schema PUBLIC di un database, utilizza il comando REVOKE per rimuovere tale autorizzazione.

Per revocare un'autorizzazione concessa precedentemente, utilizza il comando REVOKE. Le autorizzazioni del proprietario dell'oggetto, ad esempio le autorizzazioni DROP, GRANT e REVOKE, sono implicite e non possono essere concesse o revocate. I proprietari degli oggetti possono revocare le proprie autorizzazioni normali, ad esempio quello di creare una tabella in sola lettura personale e per altri utenti. Gli utenti con privilegi avanzati mantengono tutte le autorizzazioni, indipendentemente dai comandi GRANT e REVOKE.

Utenti con privilegi avanzati

Gli utenti con privilegi avanzati del database dispongono delle stesse autorizzazioni dei proprietari dei database per tutti i database.

L'utente admin, ovvero l'utente creato quando è stato avviato il cluster, è un utente con privilegi avanzati.

Per poter creare un utente con privilegi avanzati, è necessario esserlo.

Le viste e le tabelle di sistema di Amazon Redshift sono visibili solo agli utenti con privilegi avanzati o a tutti gli utenti. Solo gli utenti con privilegi avanzati possono eseguire query sulle tabelle di sistema e le visualizzazioni di sistema designate come "visibili agli utenti con privilegi avanzati". Per informazioni, consultare Tabelle e viste di sistema.

Gli utenti con privilegi avanzati possono visualizzare tutte le tabelle di catalogo. Per informazioni, consultare Tabelle di catalogo di sistema.

Un utente con privilegi avanzati di database ignora tutte le verifiche delle autorizzazioni. Gli utenti con privilegi avanzati mantengono tutte le autorizzazioni, indipendentemente dai comandi GRANT e REVOKE. Presta attenzione quando utilizzi il ruolo di un utente con privilegi avanzati. Consigliamo di svolgere la maggior parte dl lavoro con un ruolo diverso dall'utente con privilegi avanzati. Puoi creare un ruolo di amministratore con autorizzazioni più restrittive. Per ulteriori informazioni sulla creazione dei ruoli, consulta Controllo accessi basato sui ruoli (RBAC).

Per creare un nuovo utente con privilegi avanzati del database, accedi al database come utente con privilegi avanzati ed esegui un comando CREATE USER o ALTER USER con l'autorizzazione CREATEUSER.

```
CREATE USER adminuser CREATEUSER PASSWORD '1234Admin';
ALTER USER adminuser CREATEUSER;
```

Per creare, modificare o eliminare un utente con privilegi avanzati, utilizzare gli stessi comandi per gestire gli utenti. Per ulteriori informazioni, consultare Creazione, modifica ed eliminazione di utenti.

Utenti

Puoi creare e gestire gli utenti del database utilizzando i comandi SQL di Amazon Redshift CREATE USER e ALTER USER. In alternativa puoi configurare il client SQL con driver JDBC o ODBC personalizzati di Amazon Redshift. Questi gestiscono il processo di creazione degli utenti del database e delle password temporanee come parte del processo di accesso al database.

I driver autenticano gli utenti del database in base all'autenticazione (IAM). AWS Identity and Access Management Se gestisci già le identità degli utenti all'esterno AWS, puoi utilizzare un provider di identità (IdP) conforme a SAML 2.0 per gestire l'accesso alle risorse di Amazon Redshift. Utilizzi un ruolo IAM per configurare il tuo IdP e consentire AWS agli utenti federati di generare credenziali di database temporanee e accedere ai database Amazon Redshift. Per ulteriori informazioni, consultare Utilizzo dell'autenticazione IAM per generare credenziali utente di database.

Gli utenti di Amazon Redshift possono essere creati ed eliminati solo da un utente con privilegi avanzati del database. Gli utenti vengono autenticati quando accedono ad Amazon Redshift. Possono possedere database e oggetti di database (ad esempio tabelle). Possono inoltre concedere autorizzazioni per tali oggetti a utenti, gruppi e schemi per controllare chi vi ha accesso. Gli utenti che dispongono dei diritti CREATE DATABASE possono creare database e concedere le relative autorizzazioni. Gli utenti con privilegi avanzati dispongono delle autorizzazioni di proprietà per tutti i database.

Creazione, modifica ed eliminazione di utenti

Gli utenti del database sono globali in un cluster di data warehouse (non per ogni singolo database).

- Per creare un utente, utilizza il comando CREA UTENTE.
- Per creare un utente con privilegi avanzati, utilizza il comando CREA UTENTE con l'opzione CREATEUSER.
- Pr rimuovere un utente esistente, utilizza il comando DROP USER.
- Per modificare un utente, ad esempio per cambiare una password, utilizza il comando ALTER USER.

- Per visualizzare un elenco di utenti, esegui una query sulla tabella del catalogo PG_USER.

```
select * from pg_user;

  usename   | usesysid | usecreatedb | usesuper | usecatupd |  passwd  | valuntil |
 useconfig
------------+----------+-------------+----------+-----------+----------+----------
+-----------
 rdsdb      |        1 | t           | t        | t         | ******** |          |
 masteruser |      100 | t           | t        | f         | ******** |          |
 dwuser     |      101 | f           | f        | f         | ******** |          |
 simpleuser |      102 | f           | f        | f         | ******** |          |
 poweruser  |      103 | f           | t        | f         | ******** |          |
 dbuser     |      104 | t           | f        | f         | ******** |          |
(6 rows)
```

Gruppi

I gruppi sono raccolte di utenti a cui è concesso qualsiasi autorizzazione associata al gruppo. Puoi utilizzare i gruppi per assegnare le autorizzazioni. Ad esempio puoi creare gruppi diversi per vendite, amministrazione e supporto fornendo agli utenti di ciascun gruppo l'accesso appropriato ai dati necessari per svolgere il proprio lavoro. Puoi concedere o revocare le autorizzazioni a livello di gruppo e queste modifiche vengono applicate a tutti i membri del gruppo, ad eccezione degli utenti con privilegi avanzati.

Per visualizzare tutti i gruppi di utenti, eseguire una query sulla tabella del catalogo di sistema PG_GROUP:

```
select * from pg_group;
```

Ad esempio, per elencare tutti gli utenti del database per gruppo, esegui il seguente comando SQL.

```
SELECT u.usesysid
,g.groname
,u.usename
FROM pg_user u
LEFT JOIN pg_group g ON u.usesysid = ANY (g.grolist)
```

Creazione, modifica ed eliminazione di gruppi

Solo un utente con privilegi avanzati può creare, modificare o eliminare gruppi.

Puoi eseguire le seguenti operazioni:

- Per creare un gruppo, utilizza il comando CREATE GROUP.
- Per aggiungere o rimuovere utenti in un gruppo esistente, utilizza il comando ALTER GROUP.
- Per eliminare un gruppo, utilizza il comando DROP GROUP. Questo comando elimina solo il gruppo, non gli utenti membri.

Esempio di controllo dell'accesso di utenti e gruppi

In questo esempio vengono creati gruppi di utenti e utenti, quindi sono fornite loro diverse autorizzazioni per un database di Amazon Redshift che si connette a un client per applicazioni Web. Nell'esempio vengono utilizzati tre gruppi di utenti: utenti regolari di un'applicazione Web, utenti esperti di un'applicazione Web e sviluppatori Web.

1. Crea il gruppo in cui verranno assegnati gli utenti. Il set di comandi seguente crea tre gruppi di utenti diversi:

```
create group webappusers;

create group webpowerusers;

create group webdevusers;
```

2. Crea diversi utenti di database con autorizzazioni differenti e aggiungili ai gruppi.
   a. Creare due utenti e aggiungerli al gruppo WEBAPPUSERS:

```
create user webappuser1 password 'webAppuser1pass'
in group webappusers;

create user webappuser2 password 'webAppuser2pass'
in group webappusers;
```

   b. Crea un utente sviluppatore Web e aggiungilo al gruppo WEBDEVUSERS:

```
create user webdevuser1 password 'webDevuser2pass'
```

```
in group webdevusers;
```

c. Crea un utente con privilegi avanzati. Questo utente disporrà dei diritti di amministratore per la creazione di altri utenti:

```
create user webappadmin  password 'webAppadminpass1'
createuser;
```

3. Creare uno schema da associare alle tabelle di database utilizzate dall'applicazione Web e concedere ai vari gruppi di utenti l'accesso a questo schema:

a. Creare lo schema WEBAPP:

```
create schema webapp;
```

b. Concedere le autorizzazioni USAGE al gruppo WEBAPPUSERS:

```
grant usage on schema webapp to group webappusers;
```

c. Concedere le autorizzazioni USAGE al gruppo WEBPOWERUSERS:

```
grant usage on schema webapp to group webpowerusers;
```

d. Concedere le autorizzazioni ALL al gruppo WEBDEVUSERS:

```
grant all on schema webapp to group webdevusers;
```

Sono stati configurati utenti e gruppi di base. A questo punto puoi modificare utenti e gruppi.

4. Ad esempio, il seguente comando modifica il parametro search_path per WEBAPPUSER1.

```
alter user webappuser1 set search_path to webapp, public;
```

Il percorso di ricerca SEARCH_PATH specifica l'ordine con cui vengono eseguite le ricerche negli schemi per oggetti di database come tabelle e funzioni quando si fa riferimento all'oggetto utilizzando un nome semplice in cui non è specificato uno schema.

5. Dopo aver creato un gruppo, è inoltre possibile aggiungervi utenti, ad esempio si può aggiungere WEBAPPUSER2 al gruppo WEBPOWERUSERS:

```
alter group webpowerusers add user webappuser2;
```

Schemi

Un database contiene uno o più schemi con nome. Ogni schema di un database include tabelle e altri tipi di oggetti con nome. Per impostazione predefinita, un database include un solo schema, denominato PUBLIC. Puoi utilizzare gli schemi per creare gruppi di oggetti del database con lo stesso nome. Gli schemi sono simili alle directory del file system, con la differenza che non possono essere nidificati.

Puoi utilizzare nomi di oggetti del database identici in schemi diversi nello stesso database senza creare conflitti. Ad esempio, una tabella denominata MYTABLE può essere inclusa sia in MY_SCHEMA sia in YOUR_SCHEMA. Gli utenti che dispongono delle autorizzazioni necessarie possono accedere agli oggetti in più schemi in un database.

Per impostazione predefinita, un oggetto viene creato all'interno del primo schema nel percorso di ricerca del database. Per informazioni, vedi Percorso di ricerca più avanti in questa sezione.

Gli schemi possono aiutare a risolvere problemi di organizzazione e di simultaneità in un ambiente multi-utente nei modi seguenti:

- Per consentire a più sviluppatori di lavorare nello stesso database senza interferire tra loro.
- Per organizzare gli oggetti del database in gruppi logici per agevolarne la gestione.
- Per consentire alle applicazioni di inserire gli oggetti in schemi separati in modo che i nomi non entrino in collisione con i nomi degli oggetti utilizzati da altre applicazioni.

Creazione, modifica ed eliminazione di schemi

Gli utenti possono creare schemi e modificare o eliminare gli schemi di cui sono proprietari.

Puoi eseguire le seguenti operazioni:

- Per creare uno schema, utilizza il comando CREATE SCHEMA.
- Per modificare il proprietario di uno schema, utilizza il comando ALTER SCHEMA.
- Per eliminare uno schema con i relativi oggetti, utilizza il comando DROP SCHEMA.
- Per creare una tabella all'intero di uno schema, crea la tabella con il formato schema_name.table_name.

Per visualizzare un elenco di tutti gli schemi, eseguire una query sulla tabella del catalogo di sistema PG_NAMESPACE.

```
select * from pg_namespace;
```

Per visualizzare un elenco delle tabelle che appartengono a uno schema, eseguire una query sulla tabella del catalogo di sistema PG_TABLE_DEF. Ad esempio, la query seguente restituisce un elenco di tabelle nello schema PG_CATALOG.

```
select distinct(tablename) from pg_table_def
where schemaname = 'pg_catalog';
```

Percorso di ricerca

Il percorso di ricerca viene definito nel parametro search_path con un elenco di nomi di schemi separati da virgola. Il percorso di ricerca specifica l'ordine con cui vengono eseguite le ricerche negli schemi quando si fa riferimento a un oggetto, come una tabella o una funzione, utilizzando un nome semplice che non include un qualificatore di schema.

Se un oggetto viene creato senza specificare uno schema di destinazione, l'oggetto viene aggiunto al primo schema elencato nel percorso di ricerca. Se sono presenti oggetti con nomi identici in schemi diversi, un nome di oggetto che non specifica uno schema farà riferimento al primo schema del percorso di ricerca che include un oggetto con quel nome.

Per modificare lo schema predefinito per la sessione corrente, utilizza il comando SET.

Per ulteriori informazioni, vedi la descrizione search_path nella documentazione di riferimento relativa alla configurazione.

Autorizzazioni basate sullo schema

Le autorizzazioni basate sullo schema sono determinate dal proprietario dello schema:

- Di default, tutti gli utenti hanno le autorizzazioni CREATE e USAGE per lo schema PUBLIC di un database. Per impedire agli utenti di creare oggetti nello schema PUBLIC di un database, utilizza il comando REVOKE per rimuovere tale autorizzazione.
- A meno che il proprietario dell'oggetto non conceda loro l'autorizzazione USAGE, gli utenti non possono accedere a nessun oggetto presente negli schemi di cui non siano proprietari.

- Se agli utenti è stato concesso l'autorizzazione CREATE per uno schema creato da un altro utente, possono creare oggetti in tale schema.

Controllo accessi basato sui ruoli (RBAC)

Utilizzando il controllo degli accessi basato sui ruoli (RBAC) per gestire le autorizzazioni del database in Amazon Redshift, è possibile semplificare la gestione delle autorizzazioni di sicurezza in Amazon Redshift. Puoi proteggere l'accesso ai dati sensibili controllando ciò che gli utenti possono fare sia a livello generale che dettagliato. Puoi anche controllare l'accesso degli utenti alle attività normalmente limitate agli utenti con privilegi avanzati. Assegnando autorizzazioni diverse a ruoli diversi e assegnandole a diversi utenti, è possibile avere un controllo più granulare dell'accesso degli utenti.

Gli utenti con un ruolo assegnato possono eseguire solo le attività specificate dal ruolo per cui dispongono dell'autorizzazione. Ad esempio, un utente a cui è assegnato il ruolo con le autorizzazioni CREATE TABLE e DROP TABLE è autorizzato solo a eseguire tali attività. Puoi controllare l'accesso degli utenti concedendo livelli di autorizzazioni di sicurezza a utenti diversi per farli accedere solo ai dati necessari per il loro lavoro.

RBAC applica il principio delle autorizzazioni minime agli utenti in base ai requisiti del loro ruolo, indipendentemente dai tipi di oggetti coinvolti. La concessione e la revoca delle autorizzazioni vengono eseguite a livello di ruolo, senza la necessità di aggiornare le autorizzazioni per i singoli oggetti di database.

Con RBAC, puoi creare ruoli autorizzati a eseguire comandi con cui richiedere autorizzazioni di utente con privilegi avanzati. Gli utenti possono eseguire questi comandi purché sia loro assegnato un ruolo che includa tali autorizzazioni. Analogamente, puoi anche creare ruoli per limitare l'accesso a determinati comandi e assegnarli a utenti con privilegi avanzati o utenti che dispongano dell'autorizzazione per i ruoli in questione.

Per informazioni sul funzionamento di Amazon Redshift RBAC, guardare il video seguente: Presentazione di controllo accessi basato sui ruoli (RBAC) in Amazon Redshift.

Gerarchia dei ruoli

I ruoli sono raccolte di autorizzazioni che puoi assegnare a un utente o a un altro ruolo. Puoi assegnare autorizzazioni di sistema o database a un ruolo. Un utente eredita le autorizzazioni dal ruolo assegnatogli.

In RBAC, gli utenti possono avere ruoli nidificati. Puoi concedere ruoli sia agli utenti che ad altri ruoli. Quando concedi un ruolo a un utente, fornisci anche tutte le autorizzazioni associate al ruolo in questione. Se concedi il ruolo r1 a un utente, gli fornisci le autorizzazioni associate al ruolo r1. L'utente ora dispone delle autorizzazioni concesse al ruolo r1 e di tutte le autorizzazioni esistenti di cui era già in possesso.

Quando concedi un ruolo (r1) a un altro ruolo (r2), autorizzi r2 con tutte le autorizzazioni da r1. Inoltre, quando concedi il ruolo r2 a un altro ruolo (r3), le autorizzazioni di r3 sono la combinazione delle autorizzazioni di r1 e r2. In base alla gerarchia dei ruoli, r2 eredita le autorizzazioni di r1. Amazon Redshift propaga le autorizzazioni con ogni autorizzazione di ruolo. La concessione di r1 a r2 e quindi r2 a r3 autorizza r3 con tutte le autorizzazioni dei tre ruoli. Pertanto, concedendo r3 a un utente, l'utente dispone di tutte le autorizzazioni dei tre ruoli.

Amazon Redshift non consente la creazione di un ciclo di autorizzazione dei ruoli. Un ciclo di autorizzazione dei ruoli si verifica quando un ruolo nidificato viene assegnato di nuovo a un ruolo precedente nella gerarchia dei ruoli, ad esempio r3 viene assegnato nuovamente a r1. Per ulteriori informazioni su come creare i ruoli e gestire le assegnazioni dei ruoli, consulta Gestione dei ruoli in RBAC.

Assegnazione del ruolo

Gli utenti con privilegi avanzati e gli utenti normali con le autorizzazioni CREATE ROLE possono utilizzare l'istruzione CREATE ROLE per creare ruoli. Gli utenti con privilegi avanzati e gli amministratori di ruolo possono utilizzare l'istruzione GRANT ROLE per concedere un ruolo ad altri. Possono utilizzare l'istruzione REVOKE ROLE per revocare un ruolo da altri e l'istruzione DROP ROLE per eliminare i ruoli. Gli amministratori dei ruoli includono i proprietari dei ruoli e gli utenti a cui è stato concesso il ruolo con l'autorizzazione ADMIN OPTION.

Solo gli utenti con privilegi avanzati o gli amministratori del ruolo possono concedere e revocare ruoli. Puoi concedere o revocare uno o più ruoli da o verso uno o più ruoli o utenti. Utilizza l'opzione WITH ADMIN OPTION nell'istruzione GRANT ROLE per fornire le opzioni di amministrazione per tutti i ruoli concessi a tutti gli assegnatari.

Amazon Redshift supporta diverse combinazioni di assegnazioni dei ruoli, come la concessione di più ruoli o la possibilità di avere più assegnatari. L'opzione WITH ADMIN OPTION si applica solo agli utenti e non ai ruoli. Analogamente, è possibile utilizzare l'opzione WITH ADMIN OPTION nell'istruzione REVOKE ROLE per revocare il ruolo e l'autorizzazione amministrativa concessi al beneficiario. Se nell'istruzione viene utilizzata l'opzione ADMIN OPTION, viene revocata dal ruolo solo l'autorizzazione amministrativa.

L'istruzione nell'esempio seguente revoca l'autorizzazione amministrativa del ruolo `sample_role2` da `user2`.

```
REVOKE ADMIN OPTION FOR sample_role2 FROM user2;
```

Per ulteriori informazioni su come creare i ruoli e gestire le assegnazioni dei ruoli, consulta Gestione dei ruoli in RBAC.

Ruoli definiti dal sistema di Amazon Redshift

Amazon Redshift offre alcuni ruoli definiti dal sistema che sono definiti con autorizzazioni specifiche. I ruoli specifici del sistema presentano il prefisso `sys:`. Solo gli utenti con accesso appropriato possono modificare i ruoli definiti dal sistema o creare ruoli definiti dal sistema personalizzati. Non è possibile utilizzare il prefisso `sys:` per un ruolo definito dal sistema personalizzato.

La tabella seguente riepiloga i ruoli e le autorizzazioni.

Nome ruolo	Descrizione		
`sys:monitor`	Questo ruolo ha le autorizzazioni per accedere alle tabelle di catalogo o di sistema.		
`sys:operator`	Questo ruolo ha le autorizzazioni per accedere alle tabelle di catalogo o di sistema, analizzare, eseguire il vacuum o annullare le query.		
`sys:dba`	Questo ruolo dispone delle autorizzazioni per creare ed eliminare schemi e tabelle e troncare tabelle. Dispone delle autorizzazioni per creare o sostituire procedure archiviate, eliminare procedure, creare o sostituire funzioni, creare o sostituire		

Nome ruolo	Descrizione		
	funzioni esterne e creare ed eliminare visualizzazioni. Inoltre, questo ruolo eredita tutte le autorizzazioni dal ruolo sys:operator.		
`sys:superuser`	Questo ruolo dispone di tutte le autorizzazioni di sistema supportate definite in Autorizzazioni di sistema per RBAC.		
`sys:secadmin`	• Questo ruolo ha le autorizzazioni per creare, modificare ed eliminare utenti e per creare, eliminare e concedere ruoli. • Questo ruolo dispone delle autorizzazioni per ATTIVARE o DISATTIVARE RLS su una relazione e delle autorizzazioni per gestire le policy RLS e DDM (CREATE, DROP, ATTACH, DETACH e ALTER). Inoltre, tieni presente che le autorizzazioni EXPLAIN RLS, IGNORE RLS e EXPLAIN MASKING sono assegnate a questo ruolo per impostazione predefinita. • Questo ruolo può avere accesso alle tabelle utente solo quando gli viene fornita esplicitamente tale autorizzazione.		

Ruoli e utenti definiti dal sistema per la condivisione dei dati

Amazon Redshift crea ruoli e utenti per uso interno che corrispondono a datashare e consumatori di datashare. Ogni nome di ruolo interno e nome utente ha il prefisso dello spazio dei nomi riservato. `ds:` Hanno il seguente formato:

Nome	Descrizione		
`ds:`*`sharei`	Un ruolo di sistema che corrispon de a un datashare.		
`ds:`*`sharei`*`_consumei`	Un utente di sistema che corrispon de a un utente di datashare.		

Viene creato un ruolo di condivisione dei dati per ogni datashare. Contiene tutte le autorizzazioni attualmente concesse al datashare. Viene creato un utente per la condivisione dei dati per ogni consumatore di un datashare. Viene concessa l'autorizzazione per un singolo ruolo di condivisione dei dati. Un consumatore aggiunto a più datashare avrà un utente per la condivisione dei dati creato per ogni datashare.

Questi utenti e ruoli sono necessari per il corretto funzionamento della condivisione dei dati. Non possono essere modificati o eliminati e non sono accessibili o utilizzati per attività eseguite dai clienti. Puoi tranquillamente ignorarli. Per ulteriori informazioni sulla condivisione dei dati, consulta Condivisione dei dati tra cluster in Amazon Redshift.

> **Note**
>
> Non puoi utilizzare il `ds:` prefisso per creare ruoli o utenti definiti dall'utente.

Autorizzazioni di sistema per RBAC

Di seguito è riportato un elenco delle autorizzazioni di sistema che puoi concedere o revocare a un ruolo.

Comando	È necessario disporre dell'autorizzazione in uno dei seguenti modi per eseguire il comando
CREATE ROLE	• Utente con privilegi avanzati. • Utenti con l'autorizzazione CREATE ROLE.
DROP ROLE	• Utente con privilegi avanzati. • Proprietario del ruolo, ovvero l'utente che ha creato il ruolo o un utente a cui è stato concesso il ruolo con l'autorizzazione WITH ADMIN OPTION.
CREA UTENTE	• Utente con privilegi avanzati. • Utenti con l'autorizzazione CREATE USER. Questi utenti non possono creare utenti con privilegi avanzati.
DROP USER	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP USER.
ALTER USER	• Utente con privilegi avanzati. • Utenti con l'autorizzazione ALTER USER. Questi utenti non possono modificare gli utenti in utenti con privilegi avanzati e viceversa. • Utente attuale che desidera modificare la propria password.
CREATE SCHEMA	• Utente con privilegi avanzati. • Utenti con l'autorizzazione CREATE SCHEMA.
DROP SCHEMA	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP SCHEMA. • Proprietario dello schema.
ALTER DEFAULT PRIVILEGES	• Utente con privilegi avanzati. • Utenti con l'autorizzazione ALTER DEFAULT PRIVILEGES. • Utenti che modificano le proprie autorizzazioni di accesso di default.

Comando	È necessario disporre dell'autorizzazione in uno dei seguenti modi per eseguire il comando		
	• Utenti che definiscono le autorizzazioni per gli schemi per cui dispongono delle autorizzazioni di accesso.		
CREATE TABLE	• Utente con privilegi avanzati. • Utenti con l'autorizzazione CREATE TABLE. • Utenti con l'autorizzazione CREATE sugli schemi.		
DROP TABLE	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP TABLE. • Il proprietario della tabella con l'autorizzazione USAGE sullo schema.		
ALTER TABLE	• Utente con privilegi avanzati. • Utenti con l'autorizzazione ALTER TABLE. • Il proprietario della tabella con l'autorizzazione USAGE sullo schema.		

CREATE OR REPLAC FUNCTIC	• Per CREATE FUNCTION:
	  • Utente con privilegi avanzati.
	  • Utenti con l'autorizzazione CREATE OR REPLACE FUNCTION.
	  • Utenti con l'autorizzazione USAGE per la lingua.
	• Per REPLACE FUNCTION:
	  • Utente con privilegi avanzati.
	  • Utenti con l'autorizzazione CREATE OR REPLACE FUNCTION.
	  • Proprietario della funzione.		

Comando	È necessario disporre dell'autorizzazione in uno dei seguenti modi per eseguire il comando		
CREATE OR REPLAC EXTERN FUNCTIC	• Utente con privilegi avanzati. • Utenti con l'autorizzazione CREATE OR REPLACE EXTERNAL FUNCTION.		
DROP FUNCTIC	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP FUNCTION. • Proprietario della funzione.		

CREATE OR REPLAC PROCED	• Per CREATE PROCEDURE:
	  • Utente con privilegi avanzati.
	  • Utenti con l'autorizzazione CREATE OR REPLACE PROCEDURE.
	  • Utenti con l'autorizzazione USAGE per la lingua.
	• Per REPLACE PROCEDURE:
	  • Utente con privilegi avanzati.
	  • Utenti con l'autorizzazione CREATE OR REPLACE PROCEDURE.
	  • Proprietario della procedura.		

DROP PROCED	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP PROCEDURE. • Proprietario della procedura.		

Comando	È necessario disporre dell'autorizzazione in uno dei seguenti modi per eseguire il comando		

CREATE OR REPLAC VIEW	• Per CREATE VIEW:
	  • Utente con privilegi avanzati.
	  • Utenti con l'autorizzazione CREATE OR REPLACE VIEW.
	  • Utenti con l'autorizzazione CREATE per gli schemi.
	• Per REPLACE VIEW:
	  • Utente con privilegi avanzati.
	  • Utenti con l'autorizzazione CREATE OR REPLACE VIEW.
	  • Proprietario della visualizzazione.		

DROP VIEW	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP VIEW. • Proprietario della visualizzazione.		
CREATE MODEL	• Utente con privilegi avanzati. • Utenti con l'autorizzazione di sistema CREATE MODEL, che dovrebbero essere in grado di leggere la relazione di CREATE MODEL. • Utenti con l'autorizzazione CREATE MODEL.		
DROP MODEL	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP MODEL. • Proprietario del modello. • Proprietario dello schema.		
CREARE DATASH	• Utente con privilegi avanzati. • Utenti con l'autorizzazione CREATE DATASHARE. • Proprietario del database.		

Comando	È necessario disporre dell'autorizzazione in uno dei seguenti modi per eseguire il comando		
ALTER DATASH	• Utente con privilegi avanzati. • Utente con l'autorizzazione ALTER DATASHARE. • Utenti con autorizzazione ALTER o ALL sull'unità di condivisi one dati. • Per aggiungere oggetti specifici a una unità di condivisione dati, questi utenti devono avere l'autorizzazione sugli oggetti. Gli utenti devono essere i proprietari degli oggetti o disporre delle autorizzazioni SELECT, USAGE o ALL per gli oggetti.		
DROP DATASH	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP DATASHARE. • Proprietario del database.		
CREATE LIBRARY	• Utente con privilegi avanzati. • Utenti con l'autorizzazione CREATE LIBRARY o con l'autoriz zazione della lingua specificata.		
DROP LIBRARY	• Utente con privilegi avanzati. • Utenti con l'autorizzazione DROP LIBRARY. • Proprietario della libreria.		
ANALYZI	• Utente con privilegi avanzati. • Utenti con l'autorizzazione ANALYZE. • Proprietario della relazione. • Proprietario del database con cui è condivisa la tabella.		
CANCEL	• Utente con privilegi avanzati che annulla la propria query. • Utente con privilegi avanzati che annulla la query di un utente. • Utenti con l'autorizzazione CANCEL che annullano la query di un utente. • Utente che annulla la propria query.		

Comando	È necessario disporre dell'autorizzazione in uno dei seguenti modi per eseguire il comando		
TRUNCATE TABLE	• Utente con privilegi avanzati. • Utenti con l'autorizzazione TRUNCATE TABLE. • Proprietario della tabella.		
VACUUM	• Utente con privilegi avanzati. • Utenti con l'autorizzazione VACUUM. • Proprietario della tabella. • Proprietario del database con cui è condivisa la tabella.		
IGNORE RLS	• Utente con privilegi avanzati. • Gli utenti all'interno del ruolo `sys:secadmin` .		
EXPLAIN RLS	• Utente con privilegi avanzati. • Gli utenti all'interno del ruolo `sys:secadmin` .		
EXPLAIN MASKING	• Utente con privilegi avanzati. • Gli utenti all'interno del ruolo `sys:secadmin` .		

Autorizzazioni per un oggetto del database

Oltre alle autorizzazioni di sistema, Amazon Redshift include autorizzazioni per gli oggetti del database che definiscono le opzioni di accesso. I privilegi includono opzioni di accesso come la possibilità di leggere i dati in tabelle e visualizzazioni, scrivere dati e creare tabelle. Per ulteriori informazioni, consulta il comando GRANT.

Utilizzando RBAC, è possibile assegnare le autorizzazioni per gli oggetti del database ai ruoli in modo analogo alle autorizzazioni di sistema. Puoi quindi assegnare ruoli agli utenti e concedere loro le autorizzazioni di sistema e database.

ALTER DEFAULT PRIVILEGES per RBAC

Usa la dichiarazione ALTER DEFAULT PRIVILEGES per definire il set di default di autorizzazioni di accesso da applicare agli oggetti creati nel futuro dall'utente specificato. Per impostazione predefinita, gli utenti possono modificare solo le proprie autorizzazioni di accesso di default. Con RBAC, puoi

definire le autorizzazioni di accesso di default per i ruoli. Per ulteriori informazioni, consulta il comando ALTER DEFAULT PRIVILEGES.

RBAC consente di assegnare le autorizzazioni per gli oggetti di database ai ruoli, in modo analogo alle autorizzazioni di sistema. Puoi quindi assegnare ruoli agli utenti e concedere loro le autorizzazioni di sistema e/o database.

Considerazioni sull'utilizzo dei ruoli in RBAC

Quando utilizzi i ruoli RBAC, tieni in considerazione quanto riportato di seguito:

- Amazon Redshift non consente cicli di autorizzazioni dei ruoli. Non puoi concedere r1 a r2 e successivamente concedere r2 a r1.
- RBAC funziona sia per gli oggetti Amazon Redshift nativi che per le tabelle Amazon Redshift Spectrum.
- In qualità di amministratore di Amazon Redshift, puoi attivare RBAC aggiornando il cluster alla patch di manutenzione più recente per iniziare.
- Solo gli utenti con privilegi avanzati e gli utenti con l'autorizzazione di sistema CREATE ROLE possono creare ruoli.
- Solo gli utenti con privilegi avanzati o gli amministratori del ruolo possono modificare o eliminare ruoli.
- Il nome di un ruolo non può essere uguale a quello di un nome utente.
- Il nome di un ruolo non può contenere caratteri non validi, come ":/\n".
- Il nome di un ruolo non può essere una parola riservata, ad esempio PUBLIC.
- Il nome del ruolo non può iniziare con il prefisso riservato per i ruoli di default, ovvero `sys:`.
- Non è possibile eliminare un ruolo con il parametro RESTRICT quando viene concesso a un altro ruolo. L'impostazione predefinita è RESTRICT. Quando si tenta di eliminare un ruolo che ha ereditato un altro ruolo, Amazon Redshift genera un errore.
- Gli utenti che non dispongono delle autorizzazioni di amministratore per un ruolo non possono concedere o revocare un ruolo.

Gestione dei ruoli in RBAC

Di seguito viene fornito un elenco dei comandi necessari per eseguire determinate azioni:

- Per creare un ruolo, utilizza il comando CREATE ROLE.

- Per rinominare un ruolo o modificare il rispettivo proprietario, utilizza il comando ALTER ROLE.
- Per eliminare un ruolo, utilizza il comando DROP ROLE.
- Per concedere un ruolo a un utente, utilizza il comando GRANT.
- Per revocare un ruolo da un utente, utilizza il comando REVOKE.
- Per concedere autorizzazioni di sistema a un ruolo, utilizza il comando GRANT.
- Per revocare autorizzazioni di sistema da un ruolo, utilizza il comando REVOKE.

Per visualizzare l'elenco dei ruoli del cluster o del gruppo di lavoro, consulta SVV_ROLES.

Tutorial: creazione di ruoli e interrogazioni con RBAC

Con RBAC, puoi creare ruoli autorizzati a eseguire comandi con cui richiedere autorizzazioni di utente con privilegi avanzati. Gli utenti possono eseguire questi comandi purché sia loro assegnato un ruolo che includa tali autorizzazioni.

In questo tutorial, utilizzi il controllo degli accessi basato sui ruoli (RBAC) per gestire le autorizzazioni in un database che crei. Quindi ti connetti al database e interroghi il database da due ruoli diversi per testare la funzionalità di RBAC.

I due ruoli creati e utilizzati per interrogare il database sono e. `sales_ro` `sales_rw` Crei il `sales_ro` ruolo e interroghi i dati come utente con il `sales_ro` ruolo. L'`sales_ro`utente può utilizzare solo il comando SELECT ma non può utilizzare il comando UPDATE. Quindi, crei il `sales_rw` ruolo e interroghi i dati come utente con il `sales_rw` ruolo. L'`sales_rw`utente può utilizzare il comando SELECT e il comando UPDATE.

Inoltre, è possibile creare ruoli per limitare l'accesso a determinati comandi e assegnare il ruolo a superutenti o utenti.

Attività

- Prerequisiti
- Fase 1: Creare un utente amministratore
- Fase 2: Impostare gli schemi
- Fase 3: Creare un utente di sola lettura
- Passaggio 4: Interroga i dati come utente di sola lettura
- Passaggio 5: Creare un utente di lettura/scrittura
- Passaggio 6: Interroga i dati come utente con il ruolo di sola lettura ereditato

- Passaggio 7: concedere le autorizzazioni di aggiornamento e inserimento per il ruolo di lettura/scrittura
- Passaggio 8: Interroga i dati come utente di lettura/scrittura
- Passaggio 9: Analizza e archivia le tabelle in un database come utente amministratore
- Passaggio 10: Tronca le tabelle come utente di lettura/scrittura

Prerequisiti

- Crea un cluster Amazon Redshift o un gruppo di lavoro serverless caricato con il database di esempio TICKIT. Per creare un gruppo di lavoro serverless, consulta Amazon Redshift Serverless. Per creare un cluster, consulta Creare un cluster Amazon Redshift di esempio. Per ulteriori informazioni sul database di esempio TICKIT, consultaDatabase di esempio.
- Accedi a un utente con autorizzazioni di superutente o amministratore di ruolo. Solo i superutenti o gli amministratori dei ruoli possono concedere o revocare i ruoli. Per ulteriori informazioni sulle autorizzazioni richieste per RBAC, vedere. Autorizzazioni di sistema per RBAC
- Rivedere le Considerazioni sull'utilizzo dei ruoli in RBAC.

Fase 1: Creare un utente amministratore

Per configurare questo tutorial, in questo passaggio devi creare un ruolo di amministratore del database e associarlo a un utente amministratore del database. È necessario creare l'amministratore del database come superutente o amministratore di ruolo.

Esegui tutte le query in Amazon https://docs.aws.amazon.com/redshift/latest/mgmt/query-editor-v2-using.html Redshift.

1. Per creare il ruolo di amministratore db_admin, usa il seguente esempio.

```
CREATE ROLE db_admin;
```

2. Per creare un utente del database denominato dbadmin, usa l'esempio seguente.

```
CREATE USER dbadmin PASSWORD 'Test12345';
```

3. Per concedere il ruolo definito dal sistema denominato sys:dba al ruolo db_admin, utilizzate l'esempio seguente. Quando viene concesso il ruolo sys:dba, l'utente dbadmin può creare schemi e tabelle. Per ulteriori informazioni, consulta Ruoli definiti dal sistema di Amazon Redshift.

Fase 2: Impostare gli schemi

In questo passaggio, ti connetti al database come amministratore del database. Quindi, si creano due schemi e si aggiungono dati ad essi.

1. Connect al database dev come utente dbadmin utilizzando l'editor di query v2. Per ulteriori informazioni sulla connessione a un database, vedere Working with query editor v2.
2. Per creare gli schemi del database di vendita e marketing, utilizzare l'esempio seguente.

```
CREATE SCHEMA sales;
CREATE SCHEMA marketing;
```

3. Per creare e inserire valori nelle tabelle dello schema di vendita, utilizzare l'esempio seguente.

```
CREATE TABLE sales.cat(
catid smallint,
catgroup varchar(10),
catname varchar(10),
catdesc varchar(50)
);
INSERT INTO sales.cat(SELECT * FROM category);

CREATE TABLE sales.dates(
dateid smallint,
caldate date,
day char(3),
week smallint,
month char(5),
qtr char(5),
year smallint,
holiday boolean
);
INSERT INTO sales.dates(SELECT * FROM date);

CREATE TABLE sales.events(
eventid integer,
venueid smallint,
catid smallint,
dateid smallint,
eventname varchar(200),
starttime timestamp
);
```

```
INSERT INTO sales.events(SELECT * FROM event);

 CREATE TABLE sales.sale(
salesid integer,
listid integer,
sellerid integer,
buyerid integer,
eventid integer,
dateid smallint,
qtysold smallint,
pricepaid decimal(8,2),
commission decimal(8,2),
saletime timestamp
);
INSERT INTO sales.sale(SELECT * FROM sales);
```

4. Per creare e inserire valori nelle tabelle dello schema di marketing, utilizzate l'esempio seguente.

```
CREATE TABLE marketing.cat(
catid smallint,
catgroup varchar(10),
catname varchar(10),
catdesc varchar(50)
);
INSERT INTO marketing.cat(SELECT * FROM category);

CREATE TABLE marketing.dates(
dateid smallint,
caldate date,
day char(3),
week smallint,
month char(5),
qtr char(5),
year smallint,
holiday boolean
);
INSERT INTO marketing.dates(SELECT * FROM date);

CREATE TABLE marketing.events(
eventid integer,
venueid smallint,
catid smallint,
dateid smallint,
eventname varchar(200),
```

```
starttime timestamp
);
INSERT INTO marketing.events(SELECT * FROM event);

CREATE TABLE marketing.sale(
marketingid integer,
listid integer,
sellerid integer,
buyerid integer,
eventid integer,
dateid smallint,
qtysold smallint,
pricepaid decimal(8,2),
commission decimal(8,2),
saletime timestamp
);
INSERT INTO marketing.sale(SELECT * FROM marketing);
```

Passaggio 3: Creare un utente di sola lettura

In questo passaggio, crei un ruolo di sola lettura e un utente salesanalyst per il ruolo di sola lettura. L'analista delle vendite deve solo accedere in sola lettura alle tabelle dello schema di vendita per svolgere il compito assegnato di trovare gli eventi che hanno portato alle commissioni più elevate.

1. Connect al database come utente dbadmin.
2. Per creare il ruolo sales_ro, utilizzate l'esempio seguente.

```
CREATE ROLE sales_ro;
```

3. Per creare l'utente salesanalyst, utilizzate l'esempio seguente.

```
CREATE USER salesanalyst PASSWORD 'Test12345';
```

4. Per concedere l'utilizzo del ruolo sales_ro e selezionare l'accesso agli oggetti dello schema di vendita, utilizzate l'esempio seguente.

```
GRANT USAGE ON SCHEMA sales TO ROLE sales_ro;
GRANT SELECT ON ALL TABLES IN SCHEMA sales TO ROLE sales_ro;
```

5. Per concedere all'utente salesanalyst il ruolo sales_ro, usa l'esempio seguente.

```
GRANT ROLE sales_ro TO salesanalyst;
```

Passaggio 4: Interroga i dati come utente di sola lettura

In questo passaggio, l'utente salesanalyst richiede i dati dallo schema di vendita. Quindi, l'utente salesanalyst tenta di aggiornare una tabella e di leggere le tabelle nello schema di marketing.

1. Connect al database come utente salesanalyst.
2. Per trovare le 10 vendite con le commissioni più alte, usa il seguente esempio.

```
SET SEARCH_PATH TO sales;
SELECT DISTINCT events.dateid, sale.commission, cat.catname
FROM sale, events, dates, cat
WHERE events.dateid=dates.dateid AND events.dateid=sale.dateid AND events.catid =
 cat.catid
ORDER BY 2 DESC LIMIT 10;

+--------+------------+----------+
| dateid | commission | catname  |
+--------+------------+----------+
|   1880 |     1893.6 | Pop      |
|   1880 |     1893.6 | Opera    |
|   1880 |     1893.6 | Plays    |
|   1880 |     1893.6 | Musicals |
|   1861 |       1500 | Plays    |
|   2003 |       1500 | Pop      |
|   1861 |       1500 | Opera    |
|   2003 |       1500 | Plays    |
|   1861 |       1500 | Musicals |
|   1861 |       1500 | Pop      |
+--------+------------+----------+
```

3. Per selezionare 10 eventi dalla tabella degli eventi nello schema di vendita, utilizzate l'esempio seguente.

```
SELECT * FROM sales.events LIMIT 10;

+---------+---------+-------+--------+--------------------+---------------------+
| eventid | venueid | catid | dateid |     eventname      |      starttime      |
+---------+---------+-------+--------+--------------------+---------------------+
```

```
|    4836 |      73 |     9 |   1871 | Soulfest           | 2008-02-14 19:30:00 |
|    5739 |      41 |     9 |   1871 | Fab Faux           | 2008-02-14 19:30:00 |
|     627 |     229 |     6 |   1872 | High Society       | 2008-02-15 14:00:00 |
|    2563 |     246 |     7 |   1872 | Hamlet             | 2008-02-15 20:00:00 |
|    7703 |      78 |     9 |   1872 | Feist              | 2008-02-15 14:00:00 |
|    7903 |      90 |     9 |   1872 | Little Big Town    | 2008-02-15 19:30:00 |
|    7925 |     101 |     9 |   1872 | Spoon              | 2008-02-15 19:00:00 |
|    8113 |      17 |     9 |   1872 | Santana            | 2008-02-15 15:00:00 |
|     463 |     303 |     8 |   1873 | Tristan und Isolde | 2008-02-16 19:00:00 |
|     613 |     236 |     6 |   1873 | Pal Joey           | 2008-02-16 15:00:00 |
+---------+---------+-------+--------+--------------------+---------------------+
```

4. Per tentare di aggiornare il nome dell'evento per eventid 1, esegui l'esempio seguente. Questo esempio genererà un errore di autorizzazione negata perché l'utente salesanalyst dispone solo delle autorizzazioni SELECT sulla tabella degli eventi dello schema di vendita. Per aggiornare la tabella degli eventi, è necessario concedere le autorizzazioni del ruolo sales_ro a UPDATE. Per ulteriori informazioni sulla concessione delle autorizzazioni per l'aggiornamento di una tabella, vedere il parametro UPDATE per. GRANT Per ulteriori informazioni sul comando UPDATE, vedere. UPDATE

```
UPDATE sales.events
SET eventname = 'Comment event'
WHERE eventid = 1;

ERROR: permission denied for relation events
```

5. Per tentare di selezionare tutto dalla tabella degli eventi nello schema di marketing, utilizzate l'esempio seguente. Questo esempio genererà un errore di autorizzazione negata perché l'utente salesanalyst dispone solo delle autorizzazioni SELECT per la tabella degli eventi nello schema di vendita. Per selezionare i dati dalla tabella degli eventi nello schema di marketing, è necessario concedere al ruolo sales_ro le autorizzazioni SELECT sulla tabella degli eventi nello schema di marketing.

```
SELECT * FROM marketing.events;

              ERROR: permission denied for schema marketing
```

Fase 5: Creare un utente di lettura e scrittura

In questa fase, il tecnico commerciale responsabile della creazione della pipeline di estrazione, trasformazione e caricamento (ETL) per l'elaborazione dei dati nello schema di vendita riceverà l'accesso in sola lettura, ma in seguito avrà accesso in lettura e scrittura per eseguire le proprie attività.

1. Connect al database come utente dbadmin.
2. Per creare il ruolo sales_rw nello schema di vendita, usa l'esempio seguente.

```
CREATE ROLE sales_rw;
```

3. Per creare l'utente salesengineer, utilizzare l'esempio seguente.

```
CREATE USER salesengineer PASSWORD 'Test12345';
```

4. Per concedere l'utilizzo del ruolo sales_rw e selezionare l'accesso agli oggetti dello schema di vendita assegnandogli il ruolo sales_ro, utilizza l'esempio seguente. Per ulteriori informazioni su come i ruoli ereditano le autorizzazioni in Amazon Redshift, consulta. Gerarchia dei ruoli

```
GRANT ROLE sales_ro TO ROLE sales_rw;
```

5. Per assegnare il ruolo sales_rw all'utente salesengineer, usa l'esempio seguente.

```
GRANT ROLE sales_rw TO salesengineer;
```

Passaggio 6: interroga i dati come utente con il ruolo di sola lettura ereditato

In questo passaggio, l'utente salesengineer tenta di aggiornare la tabella degli eventi prima di ottenere le autorizzazioni di lettura.

1. Connect al database come utente salesengineer.
2. L'utente salesengineer può leggere correttamente i dati dalla tabella degli eventi dello schema di vendita. Per selezionare l'evento con eventid 1 dalla tabella degli eventi nello schema di vendita, utilizzate l'esempio seguente.

```
SELECT * FROM sales.events where eventid=1;
```

```
+---------+---------+-------+--------+-----------------+---------------------+
| eventid | venueid | catid | dateid |    eventname    |      starttime      |
+---------+---------+-------+--------+-----------------+---------------------+
|       1 |     305 |     8 |   1851 | Gotterdammerung | 2008-01-25 14:30:00 |
+---------+---------+-------+--------+-----------------+---------------------+
```

3. Per tentare di selezionare tutto dalla tabella degli eventi nello schema di marketing, utilizzate l'esempio seguente. L'utente salesengineer non dispone delle autorizzazioni per le tabelle dello schema di marketing, pertanto questa query genererà un errore di autorizzazione negata. Per selezionare i dati dalla tabella degli eventi nello schema di marketing, è necessario concedere al ruolo sales_rw le autorizzazioni SELECT sulla tabella degli eventi nello schema di marketing.

```
SELECT * FROM marketing.events;

ERROR: permission denied for schema marketing
```

4. Per tentare di aggiornare il nome dell'evento per eventid 1, esegui l'esempio seguente. Questo esempio genererà un errore di autorizzazione negata perché l'utente salesengineer dispone solo delle autorizzazioni di selezione sulla tabella degli eventi nello schema di vendita. Per aggiornare la tabella degli eventi, è necessario concedere le autorizzazioni del ruolo sales_rw a UPDATE.

```
UPDATE sales.events
SET eventname = 'Comment event'
WHERE eventid = 1;

ERROR: permission denied for relation events
```

Passaggio 7: concedere le autorizzazioni di aggiornamento e inserimento al ruolo di lettura-scrittura

In questo passaggio, concedi le autorizzazioni di aggiornamento e inserimento al ruolo sales_rw.

1. Connect al database come utente dbadmin.
2. Per concedere le autorizzazioni UPDATE, INSERT e DELETE al ruolo sales_rw, utilizzate l'esempio seguente.

```
GRANT UPDATE, INSERT, ON ALL TABLES IN SCHEMA sales TO role sales_rw;
```

Passaggio 8: Interroga i dati come utente di lettura/scrittura

In questo passaggio, il tecnico di vendita aggiorna correttamente la tabella dopo che al suo ruolo sono state concesse le autorizzazioni di inserimento e aggiornamento. Successivamente, il tecnico di vendita tenta di analizzare e cancellare la tabella degli eventi, ma non riesce a farlo.

1. Connect al database come utente salesengineer.
2. Per aggiornare il nome dell'evento per eventid 1, esegui l'esempio seguente.

```
UPDATE sales.events
SET eventname = 'Comment event'
WHERE eventid = 1;
```

3. Per visualizzare la modifica apportata nella query precedente, utilizzate l'esempio seguente per selezionare l'evento con eventid 1 dalla tabella degli eventi nello schema di vendita.

```
SELECT * FROM sales.events WHERE eventid=1;

+---------+---------+-------+--------+--------------+---------------------+
| eventid | venueid | catid | dateid |   eventname  |      starttime      |
+---------+---------+-------+--------+--------------+---------------------+
|       1 |     305 |     8 |   1851 | Comment event | 2008-01-25 14:30:00 |
+---------+---------+-------+--------+--------------+---------------------+
```

4. Per analizzare la tabella degli eventi aggiornata nello schema di vendita, utilizzate l'esempio seguente. Questo esempio genererà un errore di autorizzazione negata perché l'utente salesengineer non dispone delle autorizzazioni necessarie e non è il proprietario della tabella degli eventi nello schema di vendita. Per analizzare la tabella degli eventi, è necessario concedere al ruolo sales_rw le autorizzazioni per ANALIZZARE utilizzando il comando GRANT. Per ulteriori informazioni sul comando ANALYZE, vedere. ANALYZE

```
ANALYZE sales.events;

               ERROR: skipping "events" --- only table or database owner can analyze
```

5. Per riepilogare la tabella degli eventi aggiornata, utilizzate l'esempio seguente. Questo esempio genererà un errore di autorizzazione negata perché l'utente salesengineer non dispone delle autorizzazioni necessarie e non è il proprietario della tabella degli eventi nello schema di vendita. Per cancellare la tabella degli eventi, è necessario concedere le autorizzazioni del ruolo sales_rw

a VACUUM utilizzando il comando GRANT. Per ulteriori informazioni sul comando VACUUM, vedere. VACUUM

```
VACUUM sales.events;

ERROR: skipping "events" --- only table or database owner can vacuum it
```

Passaggio 9: Analizza e archivia le tabelle in un database come utente amministratore

In questo passaggio, l'utente dbadmin analizza e scarta tutte le tabelle. L'utente dispone delle autorizzazioni di amministratore su questo database, quindi è in grado di eseguire questi comandi.

1. Connect al database come utente dbadmin.
2. Per analizzare la tabella degli eventi nello schema di vendita, utilizzare l'esempio seguente.

```
ANALYZE sales.events;
```

3. Per archiviare la tabella degli eventi nello schema di vendita, utilizzate l'esempio seguente.

```
VACUUM sales.events;
```

4. Per analizzare la tabella degli eventi nello schema di marketing, usa l'esempio seguente.

```
ANALYZE marketing.events;
```

5. Per riepilogare la tabella degli eventi nello schema di marketing, usa l'esempio seguente.

```
VACUUM marketing.events;
```

Fase 10: Tronca le tabelle come utente di lettura/scrittura

In questo passaggio, l'utente salesengineer tenta di troncare la tabella degli eventi nello schema di vendita, ma riesce solo se l'utente dbadmin concede le autorizzazioni di troncamento.

1. Connect al database come utente salesengineer.
2. Per provare a eliminare tutte le righe dalla tabella degli eventi nello schema di vendita, utilizzate l'esempio seguente. Questo esempio genererà un errore perché l'utente salesengineer non dispone delle autorizzazioni necessarie e non è il proprietario della tabella degli eventi nello

schema di vendita. Per troncare la tabella degli eventi, è necessario concedere le autorizzazioni del ruolo sales_rw a TRUNCATE utilizzando il comando GRANT. Per ulteriori informazioni sul comando TRUNCATE, consulta TRUNCATE.

```
TRUNCATE sales.events;

ERROR: must be owner of relation events
```

3. Connect al database come utente dbadmin.
4. Per concedere i privilegi di troncamento della tabella al ruolo sales_rw, utilizzate l'esempio seguente.

```
GRANT TRUNCATE TABLE TO role sales_rw;
```

5. Connect al database come utente salesengineer utilizzando l'editor di query v2.
6. Per leggere i primi 10 eventi dalla tabella degli eventi nello schema di vendita, utilizzate l'esempio seguente.

```
SELECT * FROM sales.events ORDER BY eventid LIMIT 10;

+---------+---------+-------+--------+----------------------------
+---------------------+
| eventid | venueid | catid | dateid |          eventname          |      starttime
    |
+---------+---------+-------+--------+----------------------------
+---------------------+
|       1 |     305 |     8 |   1851 | Comment event               | 2008-01-25
 14:30:00 |
|       2 |     306 |     8 |   2114 | Boris Godunov               | 2008-10-15
 20:00:00 |
|       3 |     302 |     8 |   1935 | Salome                      | 2008-04-19
 14:30:00 |
|       4 |     309 |     8 |   2090 | La Cenerentola (Cinderella) | 2008-09-21
 14:30:00 |
|       5 |     302 |     8 |   1982 | Il Trovatore                | 2008-06-05
 19:00:00 |
|       6 |     308 |     8 |   2109 | L Elisir d Amore            | 2008-10-10
 19:30:00 |
|       7 |     309 |     8 |   1891 | Doctor Atomic               | 2008-03-06
 14:00:00 |
```

```
|       8 |     302 |     8 |   1832 | The Magic Flute             | 2008-01-06
 20:00:00 |
|       9 |     308 |     8 |   2087 | The Fly                     | 2008-09-18
 19:30:00 |
|      10 |     305 |     8 |   2079 | Rigoletto                   | 2008-09-10
 15:00:00 |
+---------+---------+-------+--------+-----------------------------
+---------------------+
```

7. Per troncare la tabella degli eventi nello schema di vendita, utilizzare l'esempio seguente.

```
TRUNCATE sales.events;
```

8. Per leggere i dati della tabella degli eventi aggiornata nello schema di vendita, utilizzare l'esempio seguente.

```
SELECT * FROM sales.events ORDER BY eventid LIMIT 10;

+---------+---------+-------+--------+-----------------------------
+---------------------+
| eventid | venueid | catid | dateid |          eventname          |      starttime
    |
+---------+---------+-------+--------+-----------------------------
+---------------------+
```

Crea ruoli di sola lettura e lettura-scrittura per lo schema di marketing (opzionale)

In questo passaggio, crei ruoli di sola lettura e lettura-scrittura per lo schema di marketing.

1. Connect al database come utente dbadmin.
2. Per creare ruoli di sola lettura e lettura-scrittura per lo schema di marketing, usa l'esempio seguente.

```
CREATE ROLE marketing_ro;

CREATE ROLE marketing_rw;

GRANT USAGE ON SCHEMA marketing TO ROLE marketing_ro, ROLE marketing_rw;

GRANT SELECT ON ALL TABLES IN SCHEMA marketing TO ROLE marketing_ro;
```

```
GRANT ROLE marketing_ro TO ROLE marketing_rw;

GRANT INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA marketing TO ROLE marketing_rw;

CREATE USER marketinganalyst PASSWORD 'Test12345';

CREATE USER marketingengineer PASSWORD 'Test12345';

GRANT ROLE marketing_ro TO marketinganalyst;

GRANT ROLE marketing_rw TO marketingengineer;
```

Funzioni di sistema per RBAC (opzionale)

Amazon Redshift offre due funzioni per fornire informazioni di sistema sull'appartenenza degli utenti e sull'appartenenza ai ruoli in gruppi o ruoli aggiuntivi: role_is_member_of e user_is_member_of. Queste funzioni sono disponibili per superutenti e utenti regolari. I superutenti possono controllare tutte le appartenenze ai ruoli. Gli utenti regolari possono verificare l'appartenenza solo ai ruoli a cui hanno ottenuto l'accesso.

Per utilizzare la funzione role_is_member_of

1. Connect al database come utente salesengineer.
2. Per verificare se il ruolo sales_rw è un membro del ruolo sales_ro, usa l'esempio seguente.

```
SELECT role_is_member_of('sales_rw', 'sales_ro');

+-------------------+
| role_is_member_of |
+-------------------+
| true              |
+-------------------+
```

3. Per verificare se il ruolo sales_ro è un membro del ruolo sales_rw, usa l'esempio seguente.

```
SELECT role_is_member_of('sales_ro', 'sales_rw');

+-------------------+
| role_is_member_of |
+-------------------+
| false             |
```

```
+-------------------+
```

Per utilizzare la funzione user_is_member_of

1. Connect al database come utente salesengineer.
2. L'esempio seguente tenta di verificare l'appartenenza dell'utente salesanalyst. Questa query genera un errore perché salesengineer non ha accesso a salesanalyst. Per eseguire correttamente questo comando, connettiti al database come utente salesanalyst e usa l'esempio.

```
SELECT user_is_member_of('salesanalyst', 'sales_ro');

ERROR
```

3. Connect al database come superutente.
4. Per verificare l'appartenenza dell'utente salesanalyst quando è connesso come superutente, usa il seguente esempio.

```
SELECT user_is_member_of('salesanalyst', 'sales_ro');

+-------------------+
| user_is_member_of |
+-------------------+
| true              |
+-------------------+
```

5. Connect al database come utente dbadmin.
6. Per verificare l'appartenenza dell'utente salesengineer, utilizzare l'esempio seguente.

```
SELECT user_is_member_of('salesengineer', 'sales_ro');

+-------------------+
| user_is_member_of |
+-------------------+
| true              |
+-------------------+

SELECT user_is_member_of('salesengineer', 'marketing_ro');

+-------------------+
| user_is_member_of |
```

```
+-------------------+
| false             |
+-------------------+

SELECT user_is_member_of('marketinganalyst', 'sales_ro');

+-------------------+
| user_is_member_of |
+-------------------+
| false             |
+-------------------+
```

Visualizzazioni di sistema per RBAC (opzionale)

Per visualizzare i ruoli, l'assegnazione dei ruoli agli utenti, la gerarchia dei ruoli e i privilegi per gli oggetti del database tramite i ruoli, utilizza le viste di sistema per Amazon Redshift. Queste visualizzazioni sono disponibili per superutenti e utenti regolari. I superutenti possono controllare tutti i dettagli del ruolo. Gli utenti normali possono controllare solo i dettagli dei ruoli a cui hanno ottenuto l'accesso.

1. Per visualizzare un elenco di utenti a cui sono esplicitamente concessi ruoli nel cluster, utilizzate l'esempio seguente.

   ```
   SELECT * FROM svv_user_grants;
   ```

2. Per visualizzare un elenco di ruoli a cui vengono concessi esplicitamente ruoli nel cluster, utilizzare l'esempio seguente.

   ```
   SELECT * FROM svv_role_grants;
   ```

Per l'elenco completo delle visualizzazioni di sistema, fare riferimento aViste SVV dei metadati.

Usa la sicurezza a livello di riga con RBAC (opzionale)

Per avere un controllo granulare degli accessi ai dati sensibili, utilizza la sicurezza a livello di riga (RLS). Per ulteriori informazioni sulla RLS, consulta Sicurezza a livello di riga.

In questa sezione, crei una politica RLS che concede `salesengineer` all'utente le autorizzazioni per visualizzare solo le righe della `cat` tabella che hanno il `catdesc` valore di Major League Baseball. Quindi interrogherai il database come utente. `salesengineer`

1. Connect al database come `salesengineer` utente.
2. Per visualizzare le prime 5 voci della `cat` tabella, utilizzate l'esempio seguente.

```
SELECT *
FROM sales.cat
ORDER BY catid ASC
LIMIT 5;

+-------+----------+---------+---------------------------------+
| catid | catgroup | catname |             catdesc             |
+-------+----------+---------+---------------------------------+
|     1 | Sports   | MLB     | Major League Baseball           |
|     2 | Sports   | NHL     | National Hockey League          |
|     3 | Sports   | NFL     | National Football League        |
|     4 | Sports   | NBA     | National Basketball Association |
|     5 | Sports   | MLS     | Major League Soccer             |
+-------+----------+---------+---------------------------------+
```

3. Connect al database come `dbadmin` utente.
4. Per creare una politica RLS per la `catdesc` colonna della `cat` tabella, utilizzate l'esempio seguente.

```
CREATE RLS POLICY policy_mlb_engineer
WITH (catdesc VARCHAR(50))
USING (catdesc = 'Major League Baseball');
```

5. Per associare la politica RLS al `sales_rw` ruolo, utilizzare l'esempio seguente.

```
ATTACH RLS POLICY policy_mlb_engineer ON sales.cat TO ROLE sales_rw;
```

6. Per modificare la tabella in modo da attivare RLS, utilizzate l'esempio seguente.

```
ALTER TABLE sales.cat ROW LEVEL SECURITY ON;
```

7. Connect al database come `salesengineer` utente.
8. Per tentare di visualizzare le prime 5 voci della `cat` tabella, utilizzate l'esempio seguente. Nota che solo le voci vengono visualizzate solo quando la `catdesc` colonna è`Major League Baseball`.

```
SELECT *
FROM sales.cat
ORDER BY catid ASC
LIMIT 5;

+-------+----------+---------+----------------------+
| catid | catgroup | catname |        catdesc       |
+-------+----------+---------+----------------------+
|     1 | Sports   | MLB     | Major League Baseball |
+-------+----------+---------+----------------------+
```

9. Connect al database come `salesanalyst` utente.
10. Per tentare di visualizzare le prime 5 voci della `cat` tabella, utilizzate l'esempio seguente. Tieni presente che non viene visualizzata alcuna voce perché viene applicata la politica di negazione predefinita di tutti.

```
SELECT *
FROM sales.cat
ORDER BY catid ASC
LIMIT 5;

+-------+----------+---------+----------------------+
| catid | catgroup | catname |        catdesc       |
+-------+----------+---------+----------------------+
```

11. Connect al database come `dbadmin` utente.
12. Per concedere l'autorizzazione IGNORE RLS al `sales_ro` ruolo, utilizzate l'esempio seguente. Ciò concede `salesanalyst` all'utente le autorizzazioni per ignorare le politiche RLS poiché è un membro del ruolo. `sales_ro`

```
GRANT IGNORE RLS TO ROLE sales_ro;
```

13. Connect al database come `salesanalyst` utente.
14. Per visualizzare le prime 5 voci della `cat` tabella, utilizzate l'esempio seguente.

```
SELECT *
FROM sales.cat
ORDER BY catid ASC
LIMIT 5;
```

```
+-------+----------+---------+--------------------------------+
| catid | catgroup | catname |             catdesc            |
+-------+----------+---------+--------------------------------+
|     1 | Sports   | MLB     | Major League Baseball          |
|     2 | Sports   | NHL     | National Hockey League         |
|     3 | Sports   | NFL     | National Football League       |
|     4 | Sports   | NBA     | National Basketball Association |
|     5 | Sports   | MLS     | Major League Soccer            |
+-------+----------+---------+--------------------------------+
```

15. Connect al database come `dbadmin` utente.
16. Per revocare l'autorizzazione IGNORE RLS dal `sales_ro` ruolo, utilizzate l'esempio seguente.

```
REVOKE IGNORE RLS FROM ROLE sales_ro;
```

17. Connect al database come `salesanalyst` utente.
18. Per tentare di visualizzare le prime 5 voci della `cat` tabella, utilizzate l'esempio seguente. Tieni presente che non viene visualizzata alcuna voce perché viene applicata la politica di negazione predefinita di tutti.

```
SELECT *
FROM sales.cat
ORDER BY catid ASC
LIMIT 5;

+-------+----------+---------+----------------------+
| catid | catgroup | catname |        catdesc       |
+-------+----------+---------+----------------------+
```

19. Connect al database come `dbadmin` utente.
20. Per scollegare la politica RLS dalla `cat` tabella, utilizzare l'esempio seguente.

```
DETACH RLS POLICY policy_mlb_engineer ON cat FROM ROLE sales_rw;
```

21. Connect al database come `salesanalyst` utente.
22. Per tentare di visualizzare le prime 5 voci della `cat` tabella, utilizzate l'esempio seguente. Tieni presente che non viene visualizzata alcuna voce perché viene applicata la politica di negazione predefinita di tutti.

```
SELECT *
```

```
FROM sales.cat
ORDER BY catid ASC
LIMIT 5;

+-------+----------+---------+---------------------------------+
| catid | catgroup | catname |             catdesc             |
+-------+----------+---------+---------------------------------+
|     1 | Sports   | MLB     | Major League Baseball           |
|     2 | Sports   | NHL     | National Hockey League          |
|     3 | Sports   | NFL     | National Football League        |
|     4 | Sports   | NBA     | National Basketball Association |
|     5 | Sports   | MLS     | Major League Soccer             |
+-------+----------+---------+---------------------------------+
```

23. Connect al database come `dbadmin` utente.
24. Per eliminare la politica RLS, utilizzare l'esempio seguente.

```
DROP RLS POLICY policy_mlb_engineer;
```

25. Per rimuovere RLS, utilizzate l'esempio seguente.

```
ALTER TABLE cat ROW LEVEL SECURITY OFF;
```

Argomenti correlati

Per ulteriori informazioni su RBAC, consultate la seguente documentazione:

- Gerarchia dei ruoli
- Assegnazione del ruolo
- Autorizzazioni per un oggetto del database
- ALTER DEFAULT PRIVILEGES per RBAC

Sicurezza a livello di riga

Utilizzando la sicurezza a livello di riga (RLS) in Amazon Redshift, puoi avere un controllo dettagliato degli accessi ai tuoi dati sensibili. Puoi decidere quali utenti o ruoli possono accedere a record di dati specifici all'interno di schemi o tabelle, in base alle policy di sicurezza definite a livello di oggetti di database. Oltre alla sicurezza a livello di colonna, in cui puoi concedere agli utenti le autorizzazioni

per un sottoinsieme di colonne, le policy RLS possono essere utilizzate per limitare ulteriormente l'accesso a determinate righe delle colonne visibili. Per ulteriori informazioni sulla sicurezza a livello di colonna, consulta Note di utilizzo per il controllo degli accessi a livello di colonna.

Quando si applicano le policy RLS sulle tabelle, puoi limitare i set di risultati restituiti quando gli utenti eseguono le query.

Durante la creazione di policy RLS, puoi specificare espressioni che stabiliscono se Amazon Redshift restituisce righe esistenti in una tabella in una query. Creando policy RLS per limitare l'accesso, non dovrai aggiungere o esternalizzare altre condizioni nelle query.

Quando si creano le policy RLS, consigliamo di creare policy semplici ed evitare istruzioni complesse. Quando definisci le policy RLS, non utilizzare un numero eccessivo di unioni di tabelle basate sulle policy.

Quando una policy fa riferimento a una tabella di ricerca, Amazon Redshift esegue l'analisi della tabella aggiuntiva oltre che della tabella che contiene la policy. La stessa query restituirà prestazioni differenti per un utente con una policy RLS collegata e un utente senza alcuna policy.

Utilizzo di policy RLS nelle istruzioni SQL

Quando si utilizzano policy RLS nelle istruzioni SQL, Amazon Redshift applica le seguenti regole:

- Per impostazione predefinita, Amazon Redshift applica le policy RLS alle istruzioni SELECT, UPDATE e DELETE.
- Per SELECT e UNLOAD, Amazon Redshift filtra le righe in base alla policy definita.
- Per UPDATE, Amazon Redshift aggiorna solo le righe visibili per te. Se una policy limita un sottoinsieme di righe in una tabella, non sarà possibile aggiornarle.
- Per DELETE, puoi eliminare solo le righe visibili per te. Se una policy limita un sottoinsieme di righe in una tabella, non sarà possibile aggiornarle. Per TRUNCATE, è comunque possibile troncare la tabella.
- Per CREATE TABLE LIKE, le tabelle create con le opzioni LIKE non erediteranno le impostazioni delle autorizzazioni dalla tabella di origine. Allo stesso modo, la tabella di destinazione non erediterà le policy RLS dalla tabella di origine.

Combinazione di più policy per utente

RLS in Amazon Redshift supporta il collegamento di più policy per utente e oggetto. Quando sono definite più policy per un utente, Amazon Redshift applica tutte le policy con la sintassi AND o OR a seconda dell'impostazione RLS CONJUNCTION TYPE per la tabella. Per ulteriori informazioni sui tipi di combinazione, consulta ALTER TABLE.

Più policy su una tabella possono essere associate alla tua utenza. Le policy possono essere collegate direttamente a te o puoi appartenere a più ruoli e i ruoli hanno policy diverse associate.

Quando più policy devono limitare l'accesso alle righe in una determinata relazione, puoi impostare RLS CONJUNCTION TYPE della relazione su AND. Analizza l'esempio seguente. Alice può vedere solo gli eventi sportivi che hanno il "catname" NBA, come specificato nella policy.

```
-- Create an analyst role and grant it to a user named Alice.
CREATE ROLE analyst;
CREATE USER alice WITH PASSWORD 'Name_is_alice_1';
GRANT ROLE analyst TO alice;

-- Create an RLS policy that only lets the user see sports.
CREATE RLS POLICY policy_sports
WITH (catgroup VARCHAR(10))
USING (catgroup = 'Sports');

-- Create an RLS policy that only lets the user see NBA.
CREATE RLS POLICY policy_nba
WITH (catname VARCHAR(10))
USING (catname = 'NBA');

-- Attach both to the analyst role.
ATTACH RLS POLICY policy_sports ON category TO ROLE analyst;
ATTACH RLS POLICY policy_nba ON category TO ROLE analyst;

-- Activate RLS on the category table with AND CONJUNCTION TYPE.
ALTER TABLE category ROW LEVEL SECURITY ON CONJUNCTION TYPE AND;

-- Change session to Alice.
SET SESSION AUTHORIZATION alice;

-- Select all from the category table.
SELECT catgroup, catname
FROM category;
```

```
 catgroup | catname
---------+---------
 Sports   | NBA
(1 row)
```

Quando più policy devono permettere agli utenti di vedere più righe in una determinata relazione, l'utente può impostare RLS CONJUNCTION TYPE della relazione su OR. Analizza l'esempio seguente. Alice può vedere solo "Concerts" e "Sports", come specificato nella policy.

```
-- Create an analyst role and grant it to a user named Alice.
CREATE ROLE analyst;
CREATE USER alice WITH PASSWORD 'Name_is_alice_1';
GRANT ROLE analyst TO alice;

-- Create an RLS policy that only lets the user see concerts.
CREATE RLS POLICY policy_concerts
WITH (catgroup VARCHAR(10))
USING (catgroup = 'Concerts');

-- Create an RLS policy that only lets the user see sports.
CREATE RLS POLICY policy_sports
WITH (catgroup VARCHAR(10))
USING (catgroup = 'Sports');

-- Attach both to the analyst role.
ATTACH RLS POLICY policy_concerts ON category TO ROLE analyst;
ATTACH RLS POLICY policy_sports ON category TO ROLE analyst;

-- Activate RLS on the category table with OR CONJUNCTION TYPE.
ALTER TABLE category ROW LEVEL SECURITY ON CONJUNCTION TYPE OR;

-- Change session to Alice.
SET SESSION AUTHORIZATION alice;

-- Select all from the category table.
SELECT catgroup, count(*)
FROM category
GROUP BY catgroup ORDER BY catgroup;

 catgroup | count
---------+-------
 Concerts |  3
```

```
 Sports   |  5
(2 rows)
```

Proprietà e gestione delle policy RLS

In qualità di superuser, amministratore della sicurezza o utente con il ruolo sys:secadmin, puoi creare, modificare o gestire tutte le policy RLS per le tabelle. A livello di oggetto, puoi attivare o disattivare la sicurezza a livello di riga senza modificare la definizione dello schema per le tabelle.

Per iniziare con la sicurezza a livello di riga, di seguito sono riportate le istruzioni SQL che è possibile utilizzare:

- Utilizza l'istruzione ALTER TABLE per attivare o disattivare RLS su una tabella. Per ulteriori informazioni, consulta ALTER TABLE.
- Utilizza l'istruzione CREATE RLS POLICY per creare una policy di sicurezza per una o più tabelle e specificare uno o più utenti o ruoli nella policy.

  Per ulteriori informazioni, consulta CREATE RLS POLICY.
- Utilizza l'istruzione ALTER RLS POLICY per modificare la policy, ad esempio cambiandone la definizione. Puoi utilizzare la stessa policy per più tabelle o viste.

  Per ulteriori informazioni, consulta ALTER RLS POLICY.
- Utilizza l'istruzione ATTACH RLS POLICY per collegare una policy a una o più relazioni, a uno o più utenti o ai ruoli.

  Per ulteriori informazioni, consulta ATTACH RLS POLICY .
- Utilizzate l'istruzione DETACH RLS POLICY per scollegare una politica da una o più relazioni, da uno o più utenti o dai ruoli.

  Per ulteriori informazioni, consulta DETACH RLS POLICY .
- Utilizza l'istruzione DROP RLS POLICY per eliminare una policy.

  Per ulteriori informazioni, consulta DROP RLS POLICY .
- Utilizza le istruzioni GRANT e REVOKE per concedere e revocare esplicitamente le autorizzazioni SELECT alle policy RLS che fanno riferimento alle tabelle di ricerca. Per ulteriori informazioni, consulta GRANT e REVOKE.

Per monitorare le policy create, sys:secadmin può visualizzare SVV_RLS_POLICY e SVV_RLS_ATTACHED_POLICY.

Per elencare le relazioni protette da RLS, sys:secadmin può visualizzare SVV_RLS_RELATION.

Per tracciare l'applicazione delle policy RLS su query che fanno riferimento a relazioni protette da RLS, un superutente, sys:operator o qualsiasi utente con l'autorizzazione di sistema ACCESS SYSTEM TABLE può visualizzare SVV_RLS_APPLIED_POLICY . Per impostazione predefinita, a sys:secadmin non sono concesse queste autorizzazioni.

Per interrogare le tabelle con le policy RLS collegate, ma senza vederle, puoi concedere l'autorizzazione IGNORE RLS a qualsiasi utente. Agli utenti che sono superuser o sys:secadmin l'autorizzazione IGNORE RLS viene concessa automaticamente. Per ulteriori informazioni, consulta GRANT.

Per spiegare i filtri delle policy RLS di una query nel piano EXPLAIN per la risoluzione dei problemi relativi alle query correlate a RLS, puoi concedere l'autorizzazione EXPLAIN RLS a qualsiasi utente. Per ulteriori informazioni, consulta GRANT e EXPLAIN.

Oggetti e principi dipendenti dalle policy

Per garantire la sicurezza delle applicazioni ed evitare che gli oggetti delle policy diventino obsoleti o non validi, Amazon Redshift non consente di eliminare o modificare oggetti a cui fanno riferimento le policy RLS.

Nell'esempio seguente viene illustrato come viene monitorata la dipendenza dello schema.

```
-- The CREATE and ATTACH policy statements for `policy_events` references some
-- target and lookup tables.
-- Target tables are tickit_event_redshift and target_schema.target_event_table.
-- Lookup table is tickit_sales_redshift.
-- Policy `policy_events` has following dependencies:
--   table tickit_sales_redshift column eventid, qtysold
--   table tickit_event_redshift column eventid
--   table target_event_table column eventid
--   schema public and target_schema
CREATE RLS POLICY policy_events
WITH (eventid INTEGER)
USING (
    eventid IN (SELECT eventid FROM tickit_sales_redshift WHERE qtysold <3)
);
```

```
ATTACH RLS POLICY policy_events ON tickit_event_redshift TO ROLE analyst;

ATTACH RLS POLICY policy_events ON target_schema.target_event_table TO ROLE consumer;
```

Di seguito sono elencate le dipendenze degli oggetti dello schema che Amazon Redshift monitora per le policy RLS.

- Quando si tiene traccia della dipendenza degli oggetti dello schema per la tabella di destinazione, Amazon Redshift segue queste regole:
  - Quando si elimina una tabella di destinazione, Amazon Redshift scollega la policy da una relazione, un utente, un ruolo o un pubblico.
  - Quando si rinomina il nome di una tabella di destinazione, non vi è alcun impatto sulle policy collegate.
  - Puoi eliminare le colonne della tabella delle policy a cui viene fatto riferimento all'interno della definizione della policy solo se elimini o scolleghi prima la policy. Ciò vale anche quando viene specificata l'opzione CASCADE. Puoi eliminare altre colonne nella tabella di destinazione.
  - Non è possibile rinominare le colonne di riferimento della tabella di destinazione. Per rinominare le colonne di riferimento, scollega prima la policy. Ciò vale anche quando viene specificata l'opzione CASCADE.
  - Non è possibile modificare il tipo di colonna, anche se si specifica l'opzione CASCADE.
- Quando si tiene traccia della dipendenza degli oggetti dello schema per la tabella di ricerca, Amazon Redshift segue queste regole:
  - Non è possibile eliminare una tabella di ricerca. Per eliminare una tabella di ricerca, è necessario eliminare prima la policy in cui si fa riferimento alla tabella di ricerca.
  - Non puoi rinominare una tabella di ricerca. Per rinominare una tabella di ricerca, è necessario eliminare prima la policy in cui si fa riferimento alla tabella di ricerca. Ciò vale anche quando viene specificata l'opzione CASCADE.
  - Non è possibile eliminare le colonne della tabella di ricerca utilizzate nella definizione della policy. Per eliminare le colonne di una tabella di ricerca utilizzata nella definizione di policy, è necessario eliminare prima la policy in cui si fa riferimento alla tabella di ricerca. Ciò si applica anche quando l'opzione CASCADE è specificata nell'istruzione ALTER TABLE DROP COLUMN. Puoi eliminare altre colonne nella tabella di ricerca.
  - Non è possibile rinominare le colonne di riferimento della tabella di ricerca. Per rinominare le colonne di riferimento, è necessario eliminare prima la policy in cui si fa riferimento alla tabella di ricerca. Ciò vale anche quando viene specificata l'opzione CASCADE.

  - Non puoi modificare il tipo della colonna a cui fai riferimento.
- Quando un utente o un ruolo viene eliminato, Amazon Redshift scollega automaticamente tutte le policy collegate all'utente o il ruolo.
- Se utilizzi l'opzione CASCADE nell'istruzione DROP SCHEMA, Amazon Redshift elimina anche le relazioni nello schema. Inoltre, elimina le relazioni in tutti gli altri schemi che dipendono dalle relazioni nello schema eliminato. Per una relazione che è una tabella di ricerca in una policy, Amazon Redshift non completa DROP SCHEMA DDL. Per qualsiasi relazione eliminata dall'istruzione DROP SCHEMA, Amazon Redshift scollega tutte le policy associate a tali relazioni.
- Puoi eliminare una funzione di ricerca (una funzione a cui si fa riferimento all'interno di una definizione di policy) solo se elimini anche la policy stessa. Ciò vale anche quando viene specificata l'opzione CASCADE.
- Quando una policy è collegata a una tabella, Amazon Redshift verifica in una policy diversa se questa tabella è una tabella di ricerca. In tal caso, Amazon Redshift non consentirà il collegamento di una policy a questa tabella.
- Durante la creazione di una policy RLS, Amazon Redshift verifica se questa tabella è una tabella di destinazione per qualsiasi altra policy RLS. In tal caso, Amazon Redshift non consentirà la creazione di una policy su questa tabella.

Considerazioni sull'utilizzo delle policy RLS

Di seguito sono riportate le considerazioni sull'utilizzo delle policy RLS:

- Amazon Redshift applica le policy RLS alle istruzioni SELECT, UPDATE e DELETE.
- Amazon Redshift non applica le policy RLS alle istruzioni INSERT, COPY, ALTER TABLE APPEND.
- La sicurezza a livello di riga funziona con la sicurezza a livello di colonna per proteggere i tuoi dati.
- Se il cluster Amazon Redshift utilizzava l'ultima versione disponibile a livello generale che supportava RLS, ma è stato eseguito il downgrade a una versione precedente, durante l'esecuzione di una query su tabelle di base con policy RLS allegate Amazon Redshift restituisce un errore. Il sys:secadmin può revocare l'accesso agli utenti a cui sono state concesse policy limitate, disattivare RLS sulle tabelle ed eliminare le policy.
- Se RLS è attivato per la relazione di origine, Amazon Redshift supporta l'istruzione ALTER TABLE APPEND per gli utenti con privilegi avanzati e gli utenti a cui è stata concessa esplicitamente l'autorizzazione di sistema IGNORE RLS o il ruolo sys:secadmin. In questo caso, puoi eseguire

l'istruzione ALTER TABLE APPEND per aggiungere righe a una tabella di destinazione spostando i dati da una tabella di origine esistente. Amazon Redshift sposta tutte le tuple dalla relazione di origine alla relazione di destinazione. Lo stato RLS della relazione di destinazione non influisce sull'istruzione ALTER TABLE APPEND.

- Per facilitare la migrazione da altri sistemi di data warehouse, puoi impostare e recuperare variabili di contesto di sessione personalizzate per una connessione specificando il nome e il valore della variabile.

  Nell'esempio seguente vengono impostate le variabili di contesto della sessione per una policy di sicurezza a livello di riga (RLS).

```
-- Set a customized context variable.
SELECT set_config('app.category', 'Concerts', FALSE);

-- Create a RLS policy using current_setting() to get the value of a customized
 context variable.
CREATE RLS POLICY policy_categories
WITH (catgroup VARCHAR(10))
USING (catgroup = current_setting('app.category', FALSE));

-- Set correct roles and attach the policy on the target table to one or more roles.
ATTACH RLS POLICY policy_categories ON tickit_category_redshift TO ROLE analyst, ROLE
 dbadmin;
```

  Per informazioni dettagliate su come impostare e recuperare variabili di contesto di sessione personalizzate, consulta SET, SET_CONFIG, MOSTRA, CURRENT_SETTING e RESET.

- La modifica dell'utente della sessione mediante SET SESSION AUTHORIZATION tra DECLARE e FETCH o tra le successive istruzioni FETCH non aggiornerà il piano già preparato in base alle policy utente definite con DECLARE. Evita di modificare l'utente della sessione quando i cursori vengono utilizzati con tabelle protette da RLS.
- Quando gli oggetti di base all'interno di un oggetto di visualizzazione sono protetti da RLS, le policy collegate all'utente che esegue la query vengono applicate ai rispettivi oggetti di base. Ciò non avviene invece con i controlli delle autorizzazioni a livello di oggetto, in cui le autorizzazioni del proprietario della vista vengono verificate tenendo conto degli oggetti di base della vista. È possibile visualizzare le relazioni protette da RLS di una query nell'output del piano EXPLAIN.
- Quando si fa riferimento a una funzione definita dall'utente (UDF) in una policy RLS (sicurezza a livello di riga) di una relazione collegata a un utente, questo deve disporre dell'autorizzazione EXECUTE sull'UDF per eseguire query sulla relazione.

- La sicurezza a livello di riga potrebbe limitare l'ottimizzazione delle query. Ti consigliamo di valutare attentamente le prestazioni delle query prima di distribuire viste protette da RLS su set di dati di grandi dimensioni.
- Le policy di sicurezza a livello di riga applicate alle viste con associazione tardiva potrebbero essere inviate in tabelle federate. Queste policy RLS potrebbero essere visibili nei log dei motori di elaborazione esterni.

Limitazioni

Di seguito sono riportate le limitazioni che si hanno quando si lavora con le policy RLS:

- Amazon Redshift supporta le istruzioni SELECT per determinate policy RLS con ricerche che hanno join complessi, ma non supporta le istruzioni UPDATE o DELETE. Nei casi con istruzioni UPDATE o DELETE, Amazon Redshift restituisce il seguente errore:

```
ERROR: One of the RLS policies on target relation is not supported in UPDATE/DELETE.
```

- Ogni qual volta si fa riferimento a una funzione definita dall'utente (UDF) in una policy RLS (sicurezza a livello di riga) di una relazione collegata a un utente, questo deve disporre dell'autorizzazione EXECUTE sull'UDF per eseguire query sulla relazione.
- Le query secondarie correlate non sono supportate. Amazon Redshift restituisce il seguente errore:

```
ERROR: RLS policy could not be rewritten.
```

- Le policy RLS non possono essere associate alle tabelle esterne e alle viste materializzate.
- Amazon Redshift non supporta le unità di condivisione dati con RLS. Se una relazione non ha RLS (sicurezza a livello di riga) disattivata per le unità di condivisione dati, la query restituisce il seguente errore sul cluster consumer:

```
RLS-protected relation "rls_protected_table" cannot be accessed via datasharing
 query.
```

- Nelle query tra database, Amazon Redshift blocca le letture nelle relazioni protette da RLS (sicurezza a livello di riga). Gli utenti con l'autorizzazione IGNORE RLS possono accedere alla relazione protetta utilizzando query tra database. Quando un utente senza l'autorizzazione IGNORE RLS accede a una relazione protetta da RLS tramite una query tra database, viene mostrato il seguente errore:

```
RLS-protected relation "rls_protected_table" cannot be accessed via cross-database
 query.
```

- ALTER RLS POLICY supporta solo la modifica di una policy RLS utilizzando la clausola USING (using_predicate_exp). Non puoi modificare una policy RLS con una clausola WITH quando si esegue ALTER RLS POLICY.
- Non è possibile eseguire query sulle relazioni con la sicurezza a livello di riga attivata se i valori di una delle seguenti opzioni di configurazione non corrispondono al valore predefinito della sessione:
  - `enable_case_sensitive_super_attribute`
  - `enable_case_sensitive_identifier`
  - `downcase_delimited_identifier`

  Prendi in considerazione la possibilità di reimpostare le opzioni di configurazione della sessione se tenti di eseguire query su una relazione con sicurezza a livello di riga e visualizzi il messaggio "RLS protected relation does not support session level config on case sensitivity being different from its default value".
- Quando il cluster o lo spazio dei nomi serverless di cui è stato effettuato il provisioning ha policy di sicurezza a livello di riga, i seguenti comandi sono bloccati per gli utenti normali:

  ```
  ALTER <current_user> SET enable_case_sensitive_super_attribute/
  enable_case_sensitive_identifier/downcase_delimited_identifier
  ```

  Quando crei policy RLS, consigliamo di modificare le impostazioni delle opzioni di configurazione predefinite per gli utenti normali in modo che corrispondano alle impostazioni delle opzioni di configurazione della sessione al momento della creazione della policy. Gli utenti con privilegi avanzati e gli utenti con il privilegio ALTER USER possono eseguire questa operazione utilizzando le impostazioni del gruppo di parametri o il comando ALTER USER. Per informazioni sui gruppi di parametri, consulta Gruppi di parametri di Amazon Redshift nella Guida alla gestione di Amazon Redshift. Per informazioni sul comando ALTER USER, consulta ALTER USER.
- Le viste e le viste con associazione tardiva con policy di sicurezza a livello di riga non possono essere sostituite dagli utenti normali con il comando CREATE VIEW. Per sostituire le viste o le viste con associazione tardiva con policy RLS, è necessario innanzitutto scollegare tutte le policy RLS associate, sostituire le viste o le viste con associazione tardiva e ricollegare le policy. Gli utenti con privilegi avanzati e gli utenti che dispongono di `sys:secadmin permission` possono

utilizzare CREATE VIEW sulle viste o sulle viste con associazione tardiva con policy RLS senza dover scollegare le policy.

- Le viste con policy di sicurezza a livello di riga non possono fare riferimento a tabelle di sistema e viste di sistema.
- Una vista con associazione tardiva a cui fa riferimento una visualizzazione normale non può essere protetta dalla RLS.
- Non è possibile accedere alle relazioni protette dalla RLS e ai dati annidati dei data lake nella stessa query.

Best practice per le prestazioni RLS

Di seguito sono riportate le best practice per garantire prestazioni migliori da Amazon Redshift su tabelle protette da RLS.

Sicurezza degli operatori e delle funzioni

Quando si eseguono query su tabelle protette da RLS, l'utilizzo di determinati operatori o funzioni può portare a un peggioramento delle prestazioni. Amazon Redshift classifica operatori e funzioni come sicuri o non sicuri per l'esecuzione di query su tabelle protette da RLS. Una funzione o un operatore è classificato come sicuro per RLS quando non ha effetti collaterali osservabili a seconda degli input. In particolare, una funzione o un operatore sicuro per RLS non può essere uno dei seguenti:

- Emette un valore di input, o qualsiasi valore dipendente dal valore di input, con o senza un messaggio di errore.
- Non riesce o restituisce errori che dipendono dal valore di input.

Gli operatori RLS non sicuri includono:

- Operatori aritmetici: +, -, /, *, %.
- Operatori di testo: LIKE e SIMILAR TO.
- Operatori cast.
- Funzioni definite dall'utente.

Utilizza la seguente istruzione SELECT per verificare la sicurezza degli operatori e delle funzioni.

```
SELECT proname, proc_is_rls_safe(oid) FROM pg_proc;
```

Amazon Redshift impone delle limitazioni sull'ordine di valutazione dei predicati utente contenenti operatori e funzioni non sicuri per RLS durante la pianificazione di query su tabelle protette da RLS. Le query che fanno riferimento a operatori o funzioni non sicure per RLS, durante l'esecuzione di query su tabelle protette da RLS possono causare una riduzione delle prestazioni. Le prestazioni possono peggiorare in modo significativo quando Amazon Redshift non è in grado di inviare predicati non sicuri per RLS nelle scansioni della tabella di base per sfruttare le chiavi di ordinamento. Per ottenere prestazioni migliori, evita le query che utilizzano predicati RLS non sicuri che sfruttano una chiave di ordinamento. Per verificare che Amazon Redshift sia in grado di disattivare operatori e funzioni, è possibile utilizzare le istruzioni EXPLAIN in combinazione con l'autorizzazione di sistema EXPLAIN RLS.

Caching dei risultati

Per ridurre il tempo di esecuzione delle query e migliorare le prestazioni del sistema, Amazon Redshift memorizza i risultati di certi tipi di query nella memoria sul nodo principale.

Amazon Redshift utilizza i risultati nella cache per una nuova query che esegue la scansione delle tabelle protette da RLS quando si verificano tutte queste condizioni per le tabelle non protette e quando si verificano tutte queste altre condizioni:

- La tabella o le visualizzazioni nella policy non sono state modificate.
- La policy non usa una funzione che deve essere valutata a ogni esecuzione, come GETDATE o CURRENT_USER.

Per prestazioni migliori, evita di utilizzare predicati di policy che non soddisfano le condizioni di cui sopra.

Per ulteriori informazioni sulla memorizzazione nella cache dei risultati in Amazon Redshift, consulta Caching dei risultati .

Policy complesse

Per prestazioni migliori, evita di utilizzare policy complesse con query secondarie che uniscono più tabelle.

Creazione, collegamento, scollegamento ed eliminazione di policy RLS

Puoi eseguire le seguenti operazioni:

- Per creare una policy RLS, utilizza il comando CREATE RLS POLICY.
- Per collegare una policy RLS su una tabella a uno o più utenti o ruoli, utilizza il comando ATTACH RLS POLICY .
- Per scollegare una policy di sicurezza a livello di riga su una tabella da uno o più utenti o ruoli, utilizza il comando DETACH RLS POLICY .
- Per eliminare una policy RLS per tutte le tabelle in tutti i database, utilizza il comando DROP RLS POLICY .

Di seguito è riportato un end-to-end esempio per illustrare come un superutente crea alcuni utenti e ruoli. Quindi, un utente con il ruolo secadmin crea, collega, scollega ed elimina le policy RLS. Questo esempio utilizza il database di esempio di tickit. Per ulteriori informazioni, consulta Caricamento dei dati da Amazon S3 ad Amazon Redshiftnella Guida alle operazioni di base di Amazon Redshift.

```
-- Create users and roles referenced in the policy statements.
CREATE ROLE analyst;
CREATE ROLE consumer;
CREATE ROLE dbadmin;
CREATE ROLE auditor;
CREATE USER bob WITH PASSWORD 'Name_is_bob_1';
CREATE USER alice WITH PASSWORD 'Name_is_alice_1';
CREATE USER joe WITH PASSWORD 'Name_is_joe_1';
CREATE USER molly WITH PASSWORD 'Name_is_molly_1';
CREATE USER bruce WITH PASSWORD 'Name_is_bruce_1';
GRANT ROLE sys:secadmin TO bob;
GRANT ROLE analyst TO alice;
GRANT ROLE consumer TO joe;
GRANT ROLE dbadmin TO molly;
GRANT ROLE auditor TO bruce;
GRANT ALL ON TABLE tickit_category_redshift TO PUBLIC;
GRANT ALL ON TABLE tickit_sales_redshift TO PUBLIC;
GRANT ALL ON TABLE tickit_event_redshift TO PUBLIC;

-- Create table and schema referenced in the policy statements.
CREATE SCHEMA target_schema;
GRANT ALL ON SCHEMA target_schema TO PUBLIC;
CREATE TABLE target_schema.target_event_table (LIKE tickit_event_redshift);
```

```
GRANT ALL ON TABLE target_schema.target_event_table TO PUBLIC;

-- Change session to analyst alice.
SET SESSION AUTHORIZATION alice;

-- Check the tuples visible to analyst alice.
-- Should contain all 3 categories.
SELECT catgroup, count(*)
FROM tickit_category_redshift
GROUP BY catgroup ORDER BY catgroup;

-- Change session to security administrator bob.
SET SESSION AUTHORIZATION bob;

CREATE RLS POLICY policy_concerts
WITH (catgroup VARCHAR(10))
USING (catgroup = 'Concerts');

SELECT poldb, polname, polalias, polatts, polqual, polenabled, polmodifiedby FROM
 svv_rls_policy WHERE poldb = CURRENT_DATABASE();

ATTACH RLS POLICY policy_concerts ON tickit_category_redshift TO ROLE analyst, ROLE
 dbadmin;

ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY ON;

SELECT * FROM svv_rls_attached_policy;

-- Change session to analyst alice.
SET SESSION AUTHORIZATION alice;

-- Check that tuples with only `Concert` category will be visible to analyst alice.
SELECT catgroup, count(*)
FROM tickit_category_redshift
GROUP BY catgroup ORDER BY catgroup;

-- Change session to consumer joe.
SET SESSION AUTHORIZATION joe;

-- Although the policy is attached to a different role, no tuples will be
-- visible to consumer joe because the default deny all policy is applied.
SELECT catgroup, count(*)
FROM tickit_category_redshift
GROUP BY catgroup ORDER BY catgroup;
```

```
-- Change session to dbadmin molly.
SET SESSION AUTHORIZATION molly;

-- Check that tuples with only `Concert` category will be visible to dbadmin molly.
SELECT catgroup, count(*)
FROM tickit_category_redshift
GROUP BY catgroup ORDER BY catgroup;

-- Check that EXPLAIN output contains RLS SecureScan to prevent disclosure of
-- sensitive information such as RLS filters.
EXPLAIN SELECT catgroup, count(*) FROM tickit_category_redshift GROUP BY catgroup ORDER
 BY catgroup;

-- Change session to security administrator bob.
SET SESSION AUTHORIZATION bob;

-- Grant IGNORE RLS permission so that RLS policies do not get applicable to role
 dbadmin.
GRANT IGNORE RLS TO ROLE dbadmin;

-- Grant EXPLAIN RLS permission so that anyone in role auditor can view complete
 EXPLAIN output.
GRANT EXPLAIN RLS TO ROLE auditor;

-- Change session to dbadmin molly.
SET SESSION AUTHORIZATION molly;

-- Check that all tuples are visible to dbadmin molly because `IGNORE RLS` is granted
 to role dbadmin.
SELECT catgroup, count(*)
FROM tickit_category_redshift
GROUP BY catgroup ORDER BY catgroup;

-- Change session to auditor bruce.
SET SESSION AUTHORIZATION bruce;

-- Check explain plan is visible to auditor bruce because `EXPLAIN RLS` is granted to
 role auditor.
EXPLAIN SELECT catgroup, count(*) FROM tickit_category_redshift GROUP BY catgroup ORDER
 BY catgroup;

-- Change session to security administrator bob.
SET SESSION AUTHORIZATION bob;
```

```
DETACH RLS POLICY policy_concerts ON tickit_category_redshift FROM ROLE analyst, ROLE
 dbadmin;

-- Change session to analyst alice.
SET SESSION AUTHORIZATION alice;

-- Check that no tuples are visible to analyst alice.
-- Although the policy is detached, no tuples will be visible to analyst alice
-- because of default deny all policy is applied if the table has RLS on.
SELECT catgroup, count(*)
FROM tickit_category_redshift
GROUP BY catgroup ORDER BY catgroup;

-- Change session to security administrator bob.
SET SESSION AUTHORIZATION bob;

CREATE RLS POLICY policy_events
WITH (eventid INTEGER) AS ev
USING (
    ev.eventid IN (SELECT eventid FROM tickit_sales_redshift WHERE qtysold <3)
);

ATTACH RLS POLICY policy_events ON tickit_event_redshift TO ROLE analyst;
ATTACH RLS POLICY policy_events ON target_schema.target_event_table TO ROLE consumer;

RESET SESSION AUTHORIZATION;

-- Can not cannot alter type of dependent column.
ALTER TABLE target_schema.target_event_table ALTER COLUMN eventid TYPE float;
ALTER TABLE tickit_event_redshift ALTER COLUMN eventid TYPE float;
ALTER TABLE tickit_sales_redshift ALTER COLUMN eventid TYPE float;
ALTER TABLE tickit_sales_redshift ALTER COLUMN qtysold TYPE float;

-- Can not cannot rename dependent column.
ALTER TABLE target_schema.target_event_table RENAME COLUMN eventid TO renamed_eventid;
ALTER TABLE tickit_event_redshift RENAME COLUMN eventid TO renamed_eventid;
ALTER TABLE tickit_sales_redshift RENAME COLUMN eventid TO renamed_eventid;
ALTER TABLE tickit_sales_redshift RENAME COLUMN qtysold TO renamed_qtysold;

-- Can not drop dependent column.
ALTER TABLE target_schema.target_event_table DROP COLUMN eventid CASCADE;
ALTER TABLE tickit_event_redshift DROP COLUMN eventid CASCADE;
ALTER TABLE tickit_sales_redshift DROP COLUMN eventid CASCADE;
```

```
ALTER TABLE tickit_sales_redshift DROP COLUMN qtysold CASCADE;

-- Can not drop lookup table.
DROP TABLE tickit_sales_redshift CASCADE;

-- Change session to security administrator bob.
SET SESSION AUTHORIZATION bob;

DROP RLS POLICY policy_concerts;
DROP RLS POLICY IF EXISTS policy_events;

ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY OFF;

RESET SESSION AUTHORIZATION;

-- Drop users and roles.
DROP USER bob;
DROP USER alice;
DROP USER joe;
DROP USER molly;
DROP USER bruce;
DROP ROLE analyst;
DROP ROLE consumer;
DROP ROLE auditor FORCE;
DROP ROLE dbadmin FORCE;
```

Sicurezza dei metadati

Come la sicurezza a livello di riga di Amazon Redshift, la sicurezza dei metadati offre un controllo più granulare sui metadati. Se la sicurezza dei metadati è abilitata per il cluster con provisioning o il gruppo di lavoro serverless, gli utenti possono visualizzare i metadati degli oggetti per i quali dispongono dell'accesso in visualizzazione. La sicurezza dei metadati ti consente di separare la visibilità in base alle proprie esigenze. Ad esempio, puoi utilizzare un singolo data warehouse per centralizzare tutta l'archiviazione di dati. Tuttavia, se archivi dati per più settori, la gestione della sicurezza può diventare problematica. Con la sicurezza dei metadati abilitata, puoi configurare la tua visibilità. Gli utenti di un settore possono avere maggiore visibilità sui propri oggetti, mentre limiti l'accesso in visualizzazione agli utenti di un altro settore. La sicurezza dei metadati supporta tutti i tipi di oggetti, come schemi, tabelle, viste, viste materializzate, stored procedure, funzioni definite dall'utente e modelli di machine learning.

Gli utenti possono visualizzare i metadati degli oggetti nelle seguenti circostanze:

- Se l'accesso agli oggetti è assegnato all'utente.
- Se l'accesso agli oggetti è assegnato a un gruppo o a un ruolo dell'utente.
- L'oggetto è pubblico.
- L'utente è il proprietario dell'oggetto del database.

Per abilitare la sicurezza dei metadati, utilizza il comando ALTER SYSTEM. Di seguito è riportata la sintassi di come utilizzare il comando ALTER SYSTEM per la sicurezza dei metadati.

```
ALTER SYSTEM SET metadata_security=[true|t|on|false|f|off];
```

Quando abiliti la sicurezza dei metadati, tutti gli utenti che dispongono delle autorizzazioni necessarie possono vedere i metadati pertinenti degli oggetti a cui hanno accesso. Se desideri che solo determinati utenti possano visualizzare i metadati di sicurezza, concedi l'autorizzazione ACCESS CATALOG a un ruolo e quindi assegna il ruolo all'utente. Per ulteriori informazioni sull'utilizzo dei ruoli per controllare meglio la sicurezza, consulta Controllo accessi basato sui ruoli.

L'esempio seguente illustra come concedere l'autorizzazione ACCESS CATALOG a un ruolo e quindi assegnare il ruolo a un utente. Per ulteriori informazioni su come concedere le autorizzazioni, consulta il comando GRANT.

```
CREATE ROLE sample_metadata_viewer;

GRANT ACCESS CATALOG TO ROLE sample_metadata_viewer;

GRANT ROLE sample_metadata_viewer to salesadmin;
```

Se preferisci utilizzare ruoli già definiti, i ruoli definiti dal sistema `operator`, `secadmin`, `dba` e `superuser` dispongono tutti delle autorizzazioni necessarie per visualizzare i metadati degli oggetti. Per impostazione predefinita, gli utenti con privilegi avanzati possono visualizzare il catalogo completo.

```
GRANT ROLE operator to sample_user;
```

Se utilizzi i ruoli per controllare la sicurezza dei metadati, hai accesso a tutte le viste e le funzioni di sistema fornite dal controllo degli accessi basato sui ruoli. Ad esempio, potete interrogare la vista SVV_ROLES per vedere tutti i ruoli. Per vedere se un utente è membro di un ruolo o di un gruppo, utilizza la funzione USER_IS_MEMBER_OF. Per l'elenco completo delle viste SVV, consulta

Viste SVV dei metadati. Per l'elenco delle funzioni di informazioni di sistema, consulta Funzioni di informazioni di sistema.

Mascheramento dinamico dei dati

Panoramica

Il mascheramento dinamico dei dati (DDM) in Amazon Redshift consente di proteggere i dati sensibili di un data warehouse. Puoi manipolare il modo in cui Amazon Redshift mostra i dati sensibili all'utente al momento di eseguire query, senza modificarli nel database. Puoi controllare l'accesso ai dati tramite policy di mascheramento che applicano regole di offuscamento personalizzate a un determinato utente o ruolo. Ciò consente di rispondere alle mutevoli esigenze di privacy senza alterare i dati sottostanti o modificare le query SQL.

Le policy di mascheramento dinamico dei dati nascondono, offuscano o eseguono la pseudonimizzazione dei dati che corrispondono a un determinato formato. Quando è collegata a una tabella, l'espressione di mascheramento viene applicata a una o più colonne. È possibile modificare ulteriormente le policy di mascheramento per applicarle solo a determinati utenti o a ruoli definiti dall'utente che è possibile creare con Controllo accessi basato sui ruoli (RBAC). Inoltre, è possibile applicare il mascheramento dinamico dei dati a livello di cella utilizzando colonne condizionali durante la creazione della policy di mascheramento. Per ulteriori informazioni sul mascheramento condizionale, consulta Applicazione condizionale del mascheramento dinamico dei dati.

È possibile applicare più policy di mascheramento con diversi livelli di offuscamento alla stessa colonna di una tabella e assegnarle a ruoli diversi. Per evitare conflitti quando si hanno ruoli diversi con policy distinte che si applicano a una colonna, è possibile impostare le priorità per ciascuna applicazione. In questo modo, è possibile controllare a quali dati può accedere un determinato utente o ruolo. Le policy di mascheramento dinamico dei dati possono oscurare parzialmente o completamente i dati o eseguirne l'hash utilizzando funzioni definite dall'utente scritte in SQL, Python o con AWS Lambda. Il mascheramento di dati mediante hash consente di applicare join a questi dati senza accedere a informazioni potenzialmente sensibili.

nd-to-end Un esempio

Di seguito è riportato un end-to-end esempio che mostra come creare e associare politiche di mascheramento a una colonna. Queste policy consentono agli utenti di accedere a una colonna e visualizzare valori diversi, a seconda del grado di offuscamento delle policy collegate ai loro ruoli.

Per eseguire questo esempio è necessario essere un utente con privilegi avanzati o avere il ruolo sys:secadmin.

Creazione di una policy di mascheramento

Per prima cosa, crea una tabella e inserisci i valori delle carte di credito.

```
--create the table
CREATE TABLE credit_cards (
  customer_id INT,
  credit_card TEXT
);

--populate the table with sample values
INSERT INTO credit_cards
VALUES
  (100, '4532993817514842'),
  (100, '4716002041425888'),
  (102, '5243112427642649'),
  (102, '6011720771834675'),
  (102, '6011378662059710'),
  (103, '373611968625635')
;

--run GRANT to grant permission to use the SELECT statement on the table
GRANT SELECT ON credit_cards TO PUBLIC;

--create two users
CREATE USER regular_user WITH PASSWORD '1234Test!';

CREATE USER analytics_user WITH PASSWORD '1234Test!';

--create the analytics_role role and grant it to analytics_user
--regular_user does not have a role
CREATE ROLE analytics_role;

GRANT ROLE analytics_role TO analytics_user;
```

Quindi, crea una policy di mascheramento da applicare al ruolo di analista.

```
--create a masking policy that fully masks the credit card number
CREATE MASKING POLICY mask_credit_card_full
WITH (credit_card VARCHAR(256))
```

```
USING ('000000XXXX0000'::TEXT);

--create a user-defined function that partially obfuscates credit card data
CREATE FUNCTION REDACT_CREDIT_CARD (credit_card TEXT)
RETURNS TEXT IMMUTABLE
AS $$
    import re
    regexp = re.compile("^([0-9]{6})[0-9]{5,6}([0-9]{4})")

    match = regexp.search(credit_card)
    if match != None:
        first = match.group(1)
        last = match.group(2)
    else:
        first = "000000"
        last = "0000"

    return "{}XXXXX{}".format(first, last)
$$ LANGUAGE plpythonu;

--create a masking policy that applies the REDACT_CREDIT_CARD function
CREATE MASKING POLICY mask_credit_card_partial
WITH (credit_card VARCHAR(256))
USING (REDACT_CREDIT_CARD(credit_card));

--confirm the masking policies using the associated system views
SELECT * FROM svv_masking_policy;

SELECT * FROM svv_attached_masking_policy;
```

Collegamento di una policy di mascheramento

Collega le policy di mascheramento alla tabella delle carte di credito.

```
--attach mask_credit_card_full to the credit card table as the default policy
--all users will see this masking policy unless a higher priority masking policy is
 attached to them or their role
ATTACH MASKING POLICY mask_credit_card_full
ON credit_cards(credit_card)
TO PUBLIC;

--attach mask_credit_card_partial to the analytics role
--users with the analytics role can see partial credit card information
```

```
ATTACH MASKING POLICY mask_credit_card_partial
ON credit_cards(credit_card)
TO ROLE analytics_role
PRIORITY 10;

--confirm the masking policies are applied to the table and role in the associated
 system view
SELECT * FROM svv_attached_masking_policy;

--confirm the full masking policy is in place for normal users by selecting from the
 credit card table as regular_user
SET SESSION AUTHORIZATION regular_user;

SELECT * FROM credit_cards;

--confirm the partial masking policy is in place for users with the analytics role by
 selecting from the credit card table as analytics_user
SET SESSION AUTHORIZATION analytics_user;

SELECT * FROM credit_cards;
```

Creazione di una politica di mascheramento

La sezione seguente mostra come modificare una politica di mascheramento dinamico dei dati.

```
--reset session authorization to the default
RESET SESSION AUTHORIZATION;

--alter the mask_credit_card_full policy
ALTER MASKING POLICY mask_credit_card_full
USING ('00000000000000'::TEXT);

--confirm the full masking policy is in place after altering the policy, and that
 results are altered from '000000XXXX0000' to '00000000000000'
SELECT * FROM credit_cards;
```

Scollegamento ed eliminazione di una policy di mascheramento

La sezione seguente mostra come scollegare ed eliminare le policy di mascheramento dei dati mediante la rimozione di tutte le policy di mascheramento dinamico dei dati dalla tabella.

```
--reset session authorization to the default
RESET SESSION AUTHORIZATION;
```

```
--detach both masking policies from the credit_cards table
DETACH MASKING POLICY mask_credit_card_full
ON credit_cards(credit_card)
FROM PUBLIC;

DETACH MASKING POLICY mask_credit_card_partial
ON credit_cards(credit_card)
FROM ROLE analytics_role;

--drop both masking policies
DROP MASKING POLICY mask_credit_card_full;

DROP MASKING POLICY mask_credit_card_partial;
```

Considerazioni relative all'utilizzo del mascheramento dinamico dei dati

Quando utilizzi il mascheramento dinamico dei dati prendi in considerazione quanto segue:

- Quando eseguono query sugli oggetti creati da tabelle, come le viste, gli utenti vedranno i risultati in base alle proprie policy di mascheramento, non alle policy dell'utente che ha creato gli oggetti. Ad esempio, un utente con il ruolo di analista che esegue query su una vista creata da un secadmin vedrebbe i risultati con le policy di mascheramento collegate al ruolo di analista.
- Per evitare che il comando EXPLAIN possa esporre filtri delle policy di mascheramento sensibili, solo gli utenti con l'autorizzazione SYS_EXPLAIN_DDM possono vedere le policy di mascheramento applicate negli output EXPLAIN. Gli utenti non dispongono per impostazione predefinita dell'autorizzazione SYS_EXPLAIN_DDM.

  Di seguito è riportata la sintassi per concedere l'autorizzazione a un ruolo.

  ```
  GRANT EXPLAIN MASKING TO ROLE rolename
  ```

  Per ulteriori informazioni sul comando EXPLAIN, consulta EXPLAIN.
- Gli utenti con ruoli diversi possono visualizzare risultati distinti in base alle condizioni di filtro o di join utilizzate. Ad esempio, l'esecuzione di un comando SELECT su una tabella utilizzando un valore di colonna specifico avrà esito negativo se all'utente che esegue il comando viene applicata una policy di mascheramento che offusca quella colonna.
- Le policy di mascheramento dinamico dei dati devono essere applicate prima di qualsiasi operazione o proiezione prevedibile. Le politiche di mascheramento possono includere:

- Operazioni costanti a basso costo, come la conversione di un valore in null
- Operazioni a costo moderato, come l'hashing HMAC
- Operazioni ad alto costo, come le chiamate a funzioni Lambda esterne definite dall'utente

  È consigliabile pertanto utilizzare espressioni di mascheramento semplici, quando possibile.
- È possibile utilizzare le policy di mascheramento dinamico dei dati (DMM) per i ruoli con policy di sicurezza a livello di riga, ma è importante notare che le policy RLS vengono applicate prima di quelle DDM. Un'espressione di mascheramento dei dati dinamici non sarà in grado di leggere una riga protetta dalla RLS. Per ulteriori informazioni sulla RLS, consulta Sicurezza a livello di riga.
- Quando utilizzi il comando COPY per copiare da parquet a tabelle di destinazione protette, è necessario specificare esplicitamente le colonne nell'istruzione COPY. Per ulteriori informazioni sulla mappatura delle colonne con COPY, consulta Opzioni di mappatura di colonne.
- Le policy DDM non possono essere collegate alle seguenti relazioni:
  - Tabelle e cataloghi di sistema
  - Tabelle esterna
  - Tabelle dell'unità di condivisione dati
  - Viste materializzate
  - Relazioni tra DB
  - Tabelle temporanee
  - Query correlate
- Le policy DDM possono contenere tabelle di ricerca. Le tabelle di ricerca possono essere presenti nella clausola USING. I seguenti tipi di relazione non possono essere utilizzati come tabelle di ricerca:
  - Tabelle e cataloghi di sistema
  - Tabelle esterna
  - Tabelle dell'unità di condivisione dati
  - Viste, viste materializzate e viste con associazione tardiva
  - Relazioni tra DB
  - Tabelle temporanee
  - Query correlate

  Di seguito è riportato un esempio di collegamento di una politica di mascheramento a una tabella di ricerca.

```
--Create a masking policy referencing a lookup table
CREATE MASKING POLICY lookup_mask_credit_card WITH (credit_card TEXT) USING (
  CASE
    WHEN
      credit_card IN (SELECT credit_card_lookup FROM credit_cards_lookup)
    THEN '000000XXXX0000'
    ELSE REDACT_CREDIT_CARD(credit_card)
    END
  );

--Provides access to the lookup table via a policy attached to a role
GRANT SELECT ON TABLE credit_cards_lookup TO MASKING POLICY lookup_mask_credit_card;
```

- Non è possibile collegare una policy di mascheramento che produrrebbe un output incompatibile con il tipo e la dimensione della colonna di destinazione. Ad esempio, non è possibile collegare a una colonna VARCHAR(10) una policy di mascheramento che restituisce una stringa lunga 12 caratteri. Amazon Redshift supporta le seguenti eccezioni:
  - Una politica di mascheramento con tipo di input INTN può essere collegata a una policy con dimensione INTM fino a M < N. Ad esempio, una policy di input BIGINT (INT8) può essere collegata a una colonna smallint (INT4).
  - Una politica di mascheramento con tipo di input NUMERIC o DECIMAL può sempre essere collegata a una colonna FLOAT.
- Le policy DDM non possono essere utilizzate con la condivisione dei dati. Se il producer di dati dell'unità di condivisione dati collega una policy DDM a una tabella nell'unità di condivisione dati, la tabella diventa inaccessibile agli utenti del consumer di dati che stanno cercando di eseguire query sulla tabella. Le tabelle con policy DDM collegate non possono essere aggiunte a un'unità di condivisione dati.
- Non è possibile eseguire query sulle relazioni che hanno policy DDM collegate se i valori di una delle seguenti opzioni di configurazione non corrispondono al valore predefinito della sessione:
  - `enable_case_sensitive_super_attribute`
  - `enable_case_sensitive_identifier`
  - `downcase_delimited_identifier`

  Prendi in considerazione la possibilità di reimpostare le opzioni di configurazione della sessione se tenti di eseguire query su una relazione con una policy DDM collegata e se visualizzi il messaggio "DDM protected relation does not support session level config on case sensitivity being different from its default value".

- Quando il cluster o lo spazio dei nomi serverless di cui è stato effettuato il provisioning ha una politica di mascheramento dei dati dinamici, i seguenti comandi sono bloccati per gli utenti normali:

  ```
  ALTER <current_user> SET enable_case_sensitive_super_attribute/
  enable_case_sensitive_identifier/downcase_delimited_identifier
  ```

  Quando crei policy DDM, ti consigliamo di modificare le impostazioni delle opzioni di configurazione predefinite per gli utenti normali in modo che corrispondano alle impostazioni delle opzioni di configurazione della sessione al momento della creazione della policy. Gli utenti con privilegi avanzati e gli utenti con il privilegio ALTER USER possono eseguire questa operazione utilizzando le impostazioni del gruppo di parametri o il comando ALTER USER. Per informazioni sui gruppi di parametri, consulta Gruppi di parametri di Amazon Redshift nella Guida alla gestione di Amazon Redshift. Per informazioni sul comando ALTER USER, consulta ALTER USER.
- Le viste e le viste con associazione tardiva con policy DDM collegate non possono essere sostituite dagli utenti normali con il comando CREATE VIEW. Per sostituire le viste o le viste con associazione tardiva con policy DDM, scollega innanzitutto le policy DDM associate, sostituisci le viste o le viste con associazione tardiva e ricollega le policy. Gli utenti con privilegi avanzati e gli utenti con l'autorizzazione `sys:secadmin` possono utilizzare CREATE VIEW sulle viste o sulle viste con associazione tardiva con policy DDM senza dover scollegare le policy.
- Le viste con policy DDM collegate non possono fare riferimento a tabelle e viste di sistema. Le viste con associazione tardiva possono fare riferimento alle tabelle e alle viste di sistema.
- Le viste con associazione tardiva con policy DDM collegate non possono fare riferimento a dati annidati dei data lake, come i documenti JSON.
- Le viste con associazione tardiva non possono avere policy DDM collegate se una vista fa riferimento alla vista con associazione tardiva.
- Le policy DDM vengono collegate alle viste con associazione tardiva in base al nome di colonna. Al momento della query, Amazon Redshift verifica che tutte le politiche di mascheramento collegate alla vista con associazione tardiva siano state applicate correttamente e che il tipo di colonna di output della vista con associazione tardiva corrisponda ai tipi presenti nelle politiche di mascheramento collegate. Se la convalida non riesce, Amazon Redshift restituisce un errore per la query.

Gestione delle policy di mascheramento dinamico dei dati

Puoi eseguire le seguenti operazioni:

- Per creare una policy DMM utilizza il comando CREATE MASKING POLICY.

  Di seguito è riportato un esempio di creazione di una policy di mascheramento utilizzando una funzione hash SHA-2.

```
CREATE MASKING POLICY hash_credit
WITH (credit_card varchar(256))
USING (sha2(credit_card + 'testSalt', 256));
```

- Per modificare una policy DMM utilizza il comando ALTER MASKING POLICY.

  Di seguito è riportato un esempio di modifica di una politica di mascheramento esistente.

```
ALTER MASKING POLICY hash_credit
USING (sha2(credit_card + 'otherTestSalt', 256));
```

- Per collegare una policy DMM su una tabella a uno o più utenti o ruoli, utilizza il comando ATTACH MASKING POLICY.

  Di seguito è riportato un esempio di collegamento di una policy di mascheramento a una coppia ruolo/colonna.

```
 ATTACH MASKING POLICY hash_credit
ON credit_cards (credit_card)
TO ROLE science_role
PRIORITY 30;
```

  La clausola PRIORITY determina quale policy di mascheramento si applica a una sessione utente quando più policy sono associate alla stessa colonna. Ad esempio, se l'utente nell'esempio precedente ha un'altra policy di mascheramento collegata alla stessa colonna delle carte di credito con una priorità di 20, la policy di science_role è quella che si applica, poiché ha una priorità più alta (30).

- Per scollegare una policy DMM su una tabella da uno o più utenti o ruoli, utilizza il comando DETACH MASKING POLICY.

  Di seguito è riportato un esempio di scollegamento di una policy di mascheramento da una coppia ruolo/colonna.

```
DETACH MASKING POLICY hash_credit
ON credit_cards(credit_card)
```

```
FROM ROLE science_role;
```

- Per eliminare una policy DDM da tutti i database, utilizza il comando DROP MASKING POLICY.

  Di seguito è riportato un esempio di eliminazione di una policy di mascheramento da tutti i database.

```
DROP MASKING POLICY hash_credit;
```

Gerarchia delle policy di mascheramento

Quando colleghi più politiche di mascheramento, considera quanto segue:

- È possibile collegare più policy di mascheramento a una singola colonna.
- Quando a una query sono applicabili più policy di mascheramento, viene applicata quella con la priorità più alta collegata a ciascuna colonna corrispondente. Analizza l'esempio seguente.

```
ATTACH MASKING POLICY partial_hash
ON credit_cards(address, credit_card)
TO ROLE analytics_role
PRIORITY 20;

ATTACH MASKING POLICY full_hash
ON credit_cards(credit_card, ssn)
TO ROLE auditor_role
PRIORITY 30;

SELECT address, credit_card, ssn
FROM credit_cards;
```

  Quando si esegue l'istruzione SELECT, un utente che ha sia il ruolo di analista che quello di revisore vede la colonna degli indirizzi con la policy di mascheramento `partial_hash` applicata. Vede le colonne delle carte di credito e dei numeri di previdenza sociale (SSN) con la politica di mascheramento `full_hash` applicata perché la politica `full_hash` ha la priorità più alta nella colonna delle carte di credito.
- Se non si specifica una priorità quando si collega una policy di mascheramento, verrà applicata la priorità predefinita (0).
- Non è possibile collegare due policy alla stessa colonna con la medesima priorità.

- Non è possibile collegare due politiche alla stessa combinazione di utente e colonna oppure ruolo e colonna.
- Quando più politiche di mascheramento sono applicabili allo stesso percorso SUPER e sono collegate allo stesso utente o ruolo, ha effetto solo l'associazione con la priorità più alta. Considera i seguenti esempi:

  Il primo esempio mostra due politiche di mascheramento collegate sullo stesso percorso e che ha effetto la politica con priorità più alta.

```
ATTACH MASKING POLICY hide_name
ON employees(col_person.name)
TO PUBLIC
PRIORITY 20;

ATTACH MASKING POLICY hide_last_name
ON employees(col_person.name.last)
TO PUBLIC
PRIORITY 30;

--Only the hide_last_name policy takes effect.
SELECT employees.col_person.name FROM employees;
```

  Il secondo esempio mostra due politiche di mascheramento collegate a percorsi diversi nello stesso oggetto SUPER, senza conflitti tra le politiche. Entrambi i collegamenti verranno applicati contemporaneamente.

```
ATTACH MASKING POLICY hide_first_name
ON employees(col_person.name.first)
TO PUBLIC
PRIORITY 20;

ATTACH MASKING POLICY hide_last_name
ON employees(col_person.name.last)
TO PUBLIC
PRIORITY 20;

--Both col_person.name.first and col_person.name.last are masked.
SELECT employees.col_person.name FROM employees;
```

Per verificare quale politica di mascheramento si applica a una combinazione utente e colonna o ruolo e colonna, gli utenti con il ruolo `sys:secadmin` possono cercare la coppia colonna/ruolo o colonna/utente nella vista di sistema SVV_ATTACHED_MASKING_POLICY. Per ulteriori informazioni, consulta Viste di sistema per il mascheramento dinamico dei dati.

Utilizzo del mascheramento dei dati dinamici con percorsi di tipo di dati SUPER

Amazon Redshift supporta l'associazione di politiche di mascheramento dei dati dinamici a percorsi di colonne di tipo SUPER. Per ulteriori informazioni sui tipi di dati SUPER, consultare Importazione e query di dati semistrutturati in Amazon Redshift.

Quando colleghi le politiche di mascheramento ai percorsi di colonne di tipo SUPER, tieni presenti le seguenti considerazioni.

- Quando colleghi una politica di mascheramento al percorso di una colonna, tale colonna deve essere definita come tipo di dati SUPER. È possibile applicare le politiche di mascheramento solo a valori scalari del percorso SUPER. Non è possibile applicare le politiche di mascheramento a strutture o array complessi.
- Puoi applicare diverse politiche di mascheramento a più valori scalari su una singola colonna SUPER, purché i percorsi SUPER non siano in conflitto. Ad esempio, i percorsi SUPER `a.b` e `a.b.c` sono in conflitto perché si trovano sullo stesso percorso, con `a.b` come padre di `a.b.c`. I percorsi SUPER `a.b.c` e `a.b.d` non sono in conflitto.
- Amazon Redshift non può verificare che i percorsi associati a una politica di mascheramento esistano nei dati e siano del tipo previsto finché la politica non viene applicata al runtime della query dell'utente. Ad esempio, quando colleghi una politica di mascheramento che maschera i valori TEXT in un percorso SUPER contenente un valore INT, Amazon Redshift tenta di eseguire il casting del tipo di valore del percorso.

  In tali situazioni, il comportamento di Amazon Redshift nel runtime dipende dalle impostazioni di configurazione per le query degli oggetti SUPER. Per impostazione predefinita, Amazon Redshift è in modalità permissiva e risolve i percorsi mancanti e i casting non validi come `NULL` per il percorso SUPER specificato. Per ulteriori informazioni sulle impostazioni di configurazione relative a SUPER, consulta Configurazioni di SUPER.
- SUPER è un tipo senza schema, il che significa che Amazon Redshift non può confermare l'esistenza del valore in un determinato percorso SUPER. Se colleghi una politica di mascheramento a un percorso SUPER che non esiste e Amazon Redshift è in modalità permissiva,

Amazon Redshift risolve il percorso in un valore NULL. Ti consigliamo di considerare il formato previsto degli oggetti SUPER e la probabilità che abbiano attributi inaspettati quando colleghi le politiche di mascheramento ai percorsi delle colonne SUPER. Se ritieni che ci possa essere uno schema imprevisto nella colonna SUPER, prendi in considerazione la possibilità di collegare le politiche di mascheramento direttamente alla colonna SUPER. È possibile utilizzare le funzioni di informazioni di tipo SUPER per controllare attributi e tipi nonché usare OBJECT_TRANSFORM per mascherare i valori. Per ulteriori informazioni sulle funzioni di informazioni di tipo SUPER, consulta Funzioni di informazioni sul tipo SUPER.

Esempi

Collegamento delle politiche di mascheramento ai percorsi SUPER

L'esempio seguente collega più politiche di mascheramento a più percorsi di tipo SUPER in una colonna.

```
CREATE TABLE employees (
    col_person SUPER
);

INSERT INTO employees
VALUES
    (
        json_parse('
            {
                "name": {
                    "first": "John",
                    "last": "Doe"
                },
                "age": 25,
                "ssn": "111-22-3333",
                "company": "Company Inc."
            }
        ')
    ),
    (
        json_parse('
            {
                "name": {
                    "first": "Jane",
                    "last": "Appleseed"
```

```
                },
                "age": 34,
                "ssn": "444-55-7777",
                "company": "Organization Org."
            }
        ')
    )
;
GRANT ALL ON ALL TABLES IN SCHEMA "public" TO PUBLIC;

-- Create the masking policies.

-- This policy converts the given name to all uppercase letters.
CREATE MASKING POLICY mask_first_name
WITH(first_name TEXT)
USING ( UPPER(first_name) );

-- This policy replaces the given name with the fixed string 'XXXX'.
CREATE MASKING POLICY mask_last_name
WITH(last_name TEXT)
USING ( 'XXXX'::TEXT );

-- This policy rounds down the given age to the nearest 10.
CREATE MASKING POLICY mask_age
WITH(age INT)
USING ( (FLOOR(age::FLOAT / 10) * 10)::INT );

-- This policy converts the first five digits of the given SSN to 'XXX-XX'.
CREATE MASKING POLICY mask_ssn
WITH(ssn TEXT)
USING ( 'XXX-XX-'::TEXT || SUBSTRING(ssn::TEXT FROM 8 FOR 4) );

-- Attach the masking policies to the employees table.
ATTACH MASKING POLICY mask_first_name
ON employees(col_person.name.first)
TO PUBLIC;

ATTACH MASKING POLICY mask_last_name
ON employees(col_person.name.last)
TO PUBLIC;

ATTACH MASKING POLICY mask_age
ON employees(col_person.age)
TO PUBLIC;
```

```
ATTACH MASKING POLICY mask_ssn
ON employees(col_person.ssn)
TO PUBLIC;

-- Verify that your masking policies are attached.
SELECT
    policy_name,
    TABLE_NAME,
    priority,
    input_columns,
    output_columns
FROM
    svv_attached_masking_policy;

   policy_name   | table_name | priority |           input_columns           |
 output_columns
-----------------+------------+----------+-----------------------------------
+-----------------------------------
 mask_age        | employees  |        0 | ["col_person.\"age\""]            |
 ["col_person.\"age\""]
 mask_first_name | employees  |        0 | ["col_person.\"name\".\"first\""] |
 ["col_person.\"name\".\"first\""]
 mask_last_name  | employees  |        0 | ["col_person.\"name\".\"last\""]  |
 ["col_person.\"name\".\"last\""]
 mask_ssn        | employees  |        0 | ["col_person.\"ssn\""]            |
 ["col_person.\"ssn\""]
(4 rows)

-- Observe the masking policies taking effect.
SELECT col_person FROM employees ORDER BY col_person.age;

-- This result is formatted for ease of reading.
         col_person
-------------------------------
{
    "name": {
        "first": "JOHN",
        "last": "XXXX"
    },
    "age": 20,
    "ssn": "XXX-XX-3333",
    "company": "Company Inc."
}
```

```
{
    "name": {
        "first": "JANE",
        "last": "XXXX"
    },
    "age": 30,
    "ssn": "XXX-XX-7777",
    "company": "Organization Org."
}
```

Di seguito sono riportati alcuni esempi di collegamento di politiche di mascheramento non valide ai percorsi SUPER.

```
-- This attachment fails because there is already a policy
-- with equal priority attached to employees.name.last, which is
-- on the same SUPER path as employees.name.
ATTACH MASKING POLICY mask_ssn
ON employees(col_person.name)
TO PUBLIC;
ERROR:  DDM policy "mask_last_name" is already attached on relation "employees" column
 "col_person."name"."last"" with same priority

-- Create a masking policy that masks DATETIME objects.
CREATE MASKING POLICY mask_date
WITH(INPUT DATETIME)
USING ( INPUT );

-- This attachment fails because SUPER type columns can't contain DATETIME objects.
ATTACH MASKING POLICY mask_date
ON employees(col_person.company)
TO PUBLIC;
ERROR:  cannot attach masking policy for output of type "timestamp without time zone"
 to column "col_person."company"" of type "super
```

Di seguito è riportato un esempio di collegamento di una politica di mascheramento a un percorso SUPER che non esiste. Per impostazione predefinita, Amazon Redshift risolve il percorso come `NULL`.

```
ATTACH MASKING POLICY mask_first_name
ON employees(col_person.not_exists)
TO PUBLIC;
```

```
SELECT col_person FROM employees LIMIT 1;

-- This result is formatted for ease of reading.
         col_person
----------------------------------
{
    "name": {
        "first": "JOHN",
        "last": "XXXX"
    },
    "age": 20,
    "ssn": "XXX-XX-3333",
    "company": "Company Inc.",
    "not_exists": null
}
```

Applicazione condizionale del mascheramento dinamico dei dati

È possibile eseguire il mascheramento dei dati a livello di cella creando policy di mascheramento con espressioni condizionali. Ad esempio, è possibile creare una policy di mascheramento che applichi maschere diverse a un valore, a seconda del valore di un'altra colonna in quella riga.

Di seguito è riportato un esempio di utilizzo di mascheramento condizionale dei dati per creare e collegare una policy di mascheramento che oscuri parzialmente i numeri delle carte di credito coinvolte in una frode e nasconda completamente i numeri di tutte le altre carte di credito. Per eseguire questo esempio è necessario essere un utente con privilegi avanzati o avere il ruolo sys:secadmin.

```
--Create an analyst role.
CREATE ROLE analyst;

--Create a credit card table. The table contains an is_fraud boolean column,
--which is TRUE if the credit card number in that row was involved in a fraudulent
 transaction.
CREATE TABLE credit_cards (id INT, is_fraud BOOLEAN, credit_card_number VARCHAR(16));

--Create a function that partially redacts credit card numbers.
CREATE FUNCTION REDACT_CREDIT_CARD (credit_card VARCHAR(16))
RETURNS VARCHAR(16) IMMUTABLE
AS $$
    import re
    regexp = re.compile("^([0-9]{6})[0-9]{5,6}([0-9]{4})")
```

```
    match = regexp.search(credit_card)
    if match != None:
        first = match.group(1)
        last = match.group(2)
    else:
        first = "000000"
        last = "0000"

    return "{}XXXXX{}".format(first, last)
$$ LANGUAGE plpythonu;

--Create a masking policy that partially redacts credit card numbers if the is_fraud
 value for that row is TRUE,
--and otherwise blanks out the credit card number completely.
CREATE MASKING POLICY card_number_conditional_mask
    WITH (fraudulent BOOLEAN, pan varchar(16))
    USING (CASE WHEN fraudulent THEN REDACT_CREDIT_CARD(pan)
                ELSE Null
           END);

--Attach the masking policy to the credit_cards/analyst table/role pair.
ATTACH MASKING POLICY card_number_conditional_mask ON credit_cards (credit_card_number)
 USING (is_fraud, credit_card_number)
 TO ROLE analyst PRIORITY 100;
```

Viste di sistema per il mascheramento dinamico dei dati

Gli utenti avanzati, gli utenti con il `sys:operator` ruolo e gli utenti con l'autorizzazione ACCESS SYSTEM TABLE possono accedere alle seguenti viste di sistema relative a DDM.

- SVV_MASKING_POLICY

  Utilizzate SVV_MASKING_POLICY per visualizzare tutte le politiche di mascheramento create nel cluster o nel gruppo di lavoro.

- SVV_ATTACHED_MASKING_POLICY

  Utilizzate SVV_ATTACHED_MASKING_POLICY per visualizzare tutte le relazioni e gli utenti o i ruoli con politiche allegate al database attualmente connesso.

- SYS_APPLIED_MASKING_POLICY_LOG

Utilizzate SYS_APPLIED_MASKING_POLICY_LOG per tracciare l'applicazione delle politiche di mascheramento sulle query che fanno riferimento a relazioni protette da DDM.

Di seguito sono riportati alcuni esempi di informazioni che è possibile trovare utilizzando le viste di sistema.

```
--Select all policies associated with specific users, as opposed to roles
SELECT policy_name,
       schema_name,
       table_name,
       grantee
FROM svv_attached_masking_policy
WHERE grantee_type = 'user';

--Select all policies attached to a specific user
SELECT policy_name,
       schema_name,
       table_name,
       grantee
FROM svv_attached_masking_policy
WHERE grantee = 'target_grantee_name'

--Select all policies attached to a given table
SELECT policy_name,
       schema_name,
       table_name,
       grantee
FROM svv_attached_masking_policy
WHERE table_name = 'target_table_name'
      AND schema_name = 'target_schema_name';

--Select the highest priority policy attachment for a given role
SELECT samp.policy_name,
       samp.priority,
       samp.grantee,
       smp.policy_expression
FROM svv_masking_policy AS smp
JOIN svv_attached_masking_policy AS samp
    ON samp.policy_name = smp.policy_name
WHERE
    samp.grantee_type = 'role' AND
    samp.policy_name = mask_get_policy_for_role_on_column(
```

```
        'target_schema_name',
        'target_table_name',
        'target_column_name',
        'target_role_name')
ORDER BY samp.priority desc
LIMIT 1;

--See which policy a specific user will see on a specific column in a given relation
SELECT samp.policy_name,
       samp.priority,
       samp.grantee,
       smp.policy_expression
FROM svv_masking_policy AS smp
JOIN svv_attached_masking_policy AS samp
    ON samp.policy_name = smp.policy_name
WHERE
    samp.grantee_type = 'role' AND
    samp.policy_name = mask_get_policy_for_user_on_column(
        'target_schema_name',
        'target_table_name',
        'target_column_name',
        'target_user_name')
ORDER BY samp.priority desc;

 --Select all policies attached to a given relation.
SELECT policy_name,
schema_name,
relation_name,
database_name
FROM sys_applied_masking_policy_log
WHERE relation_name = 'relation_name'
AND schema_name = 'schema_name';
```

Autorizzazioni con ambito

Le autorizzazioni con ambito consentono di concedere autorizzazioni a un utente o a un ruolo su tutti gli oggetti di un tipo all'interno di un database o di uno schema. Gli utenti e i ruoli con autorizzazioni mirate dispongono delle autorizzazioni specificate su tutti gli oggetti attuali e futuri all'interno del database o dello schema.

Per ulteriori informazioni sull'applicazione delle autorizzazioni con ambito, consulta GRANT e REVOKE.

Considerazioni sull'utilizzo delle autorizzazioni con ambito

Quando usi le autorizzazioni con ambito, tieni in considerazione quanto segue:

- È possibile utilizzare le autorizzazioni con ambito per concedere o revocare le autorizzazioni su un database o un ambito dello schema a o da un utente o un ruolo specificato.
- Non è possibile concedere autorizzazioni con ambito ai gruppi di utenti.
- L'assegnazione o la revoca delle autorizzazioni con ambito modifica le autorizzazioni per tutti gli oggetti attuali e futuri dell'ambito.
- Le autorizzazioni con ambito e le autorizzazioni a livello di oggetto funzionano indipendentemente l'una dall'altra. Ad esempio, un utente manterrà le autorizzazioni su una tabella in entrambi i casi seguenti.
  - All'utente viene concessa l'autorizzazione SELECT sulla tabella schema1.table1 e l'autorizzazione SELECT scoped su schema1. All'utente viene quindi revocata la funzione SELECT per tutte le tabelle dello schema schema1. L'utente mantiene SELECT su schema1.table1.
  - All'utente viene concessa l'autorizzazione SELECT sulla tabella schema1.table1 e l'autorizzazione SELECT scoped su schema1. All'utente viene quindi revocata la funzione SELECT per schema1.table1. L'utente mantiene SELECT su schema1.table1.
- Per assegnare o revocare le autorizzazioni con ambito è necessario soddisfare uno dei seguenti criteri:
  - Utenti con privilegi avanzati.
  - Utenti con l'opzione di assegnazione per l'autorizzazione. Per ulteriori informazioni sulle opzioni di concessione, vai al parametro WITH GRANT OPTION in. GRANT
- Le autorizzazioni con ambito possono essere assegnate o revocate solo agli oggetti del database connesso o ai database importati da un'unità di condivisione dati.
- È possibile utilizzare le autorizzazioni con ambito per impostare le autorizzazioni predefinite su un database creato da un datashare. Un utente dell'unità di condivisione dati sul lato consumer a cui sono concesse le autorizzazioni con ambito su un database condiviso ottiene automaticamente le autorizzazioni per qualsiasi nuovo oggetto aggiunto all'unità di condivisione dati sul lato producer.
- I produttori possono concedere autorizzazioni mirate sugli oggetti all'interno di uno schema a un datashare. (anteprima)

Documentazione di riferimento a SQL

Argomenti

SQL Amazon Redshift

Argomenti

Amazon Redshift è basato sull'SQL standard del settore, con funzionalità aggiunta per gestire set di dati molto grandi e supportare l'analisi e la creazione di report a elevate prestazioni su tali dati.

> **Note**
>
> La dimensione massima per una istruzione SQL di Amazon Redshift è 16 MB.

Funzioni SQL supportate sul nodo principale

Alcune query di Amazon Redshift vengono distribuite ed eseguite sui nodi di calcolo mentre altre vengono eseguite esclusivamente sul nodo principale.

Il nodo principale distribuisce SQL ai nodi di calcolo ogni volta che una query fa riferimento a tabelle create dall'utente o a tabelle di sistema (tabelle con prefisso STL o STV e visualizzazioni di sistema con un prefisso SVL o SVV). Una query che fa riferimento solo a tabelle di catalogo (tabelle con un prefisso PG, ad esempio PG_TABLE_DEF, che risiedono solo sul nodo principale) o che non fa riferimento ad alcuna tabella viene eseguita soltanto sul nodo principale.

Alcune funzioni SQL di Amazon Redshift sono supportate solo sul nodo principale e non sui nodi di calcolo. Una query che utilizza una funzione del nodo principale deve essere eseguita esclusivamente sul nodo principale e non sui nodi di calcolo, altrimenti verrà restituito un errore.

La documentazione per ciascuna funzione che deve essere eseguita esclusivamente sul nodo principale comprende una nota che indica che la funzione restituirà un errore nel caso in cui faccia riferimento a tabelle definite dall'utente o a tabelle di sistema di Amazon Redshift. consultare Nodo principale: solo funzioni per un elenco delle funzioni eseguite esclusivamente sul nodo principale.

Esempi

I seguenti esempi utilizzano il database TICKIT di esempio. Per ulteriori informazioni sul database di esempio, vai aDatabase di esempio.

CURRENT_SCHEMA

La funzione CURRENT_SCHEMA è una funzione esclusivamente per il nodo principale. In questo esempio, la query non fa riferimento a una tabella, pertanto è in esecuzione solo sul nodo principale.

```
select current_schema();

current_schema
---------------
public
```

Nell'esempio successivo, la query fa riferimento a una tabella di catalogo di sistema, pertanto è in esecuzione solo sul nodo principale.

```
select * from pg_table_def
where schemaname = current_schema() limit 1;

 schemaname | tablename | column |   type   | encoding | distkey | sortkey | notnull
------------+-----------+--------+----------+----------+---------+---------+---------
 public     | category  | catid  | smallint | none     | t       |       1 | t
```

Nell'esempio successivo, la query fa riferimento a una tabella di sistema di Amazon Redshift che si trova sui nodi di calcolo, quindi restituisce un errore.

```
select current_schema(), userid from users;

INFO:  Function "current_schema()" not supported.
```

```
ERROR:  Specified types or functions (one per INFO message) not supported on Amazon
 Redshift tables.
```

SUBSTR

SUBSTR è anche una funzione solo per nodo principale. Nell'esempio seguente la query viene eseguita in modo esclusivo sul nodo principale perché non fa riferimento a una tabella.

```
SELECT SUBSTR('amazon', 5);

+--------+
| substr |
+--------+
| on     |
+--------+
```

Nell'esempio seguente, la query fa riferimento a una tabella che si trova nei nodi di calcolo. Ciò comporta la generazione di un errore.

```
SELECT SUBSTR(catdesc, 1) FROM category LIMIT 1;

ERROR: SUBSTR() function is not supported (Hint: use SUBSTRING instead)
```

Per eseguire correttamente la query precedente, utilizza SUBSTRING.

```
SELECT SUBSTRING(catdesc, 1) FROM category LIMIT 1;

+---------------------------------+
|            substring            |
+---------------------------------+
| National Basketball Association |
+---------------------------------+
```

FATTORIALE ()

FACTORIAL () è una funzione solo per il nodo leader. Nell'esempio seguente la query viene eseguita in modo esclusivo sul nodo principale perché non fa riferimento a una tabella.

```
SELECT FACTORIAL(5);

 factorial
```

```
-------------
 120
```

Nell'esempio seguente, la query fa riferimento a una tabella che si trova nei nodi di calcolo. Ciò comporta un errore quando viene eseguito utilizzando l'editor di query v2.

```
create table t(a int);
insert into t values (5);
select factorial(a) from t;

ERROR: Specified types or functions (one per INFO message) not supported on Redshift
 tables.
Info: Function "factorial(bigint)" not supported.
```

Amazon Redshift e PostgreSQL

Argomenti

Amazon Redshift è basato su PostgreSQL. Amazon Redshift e PostgreSQL si differenziano per un certo numero di aspetti molto importanti, di cui bisogna tener conto mentre si progettano e sviluppano le applicazioni di data warehouse.

Amazon Redshift è appositamente progettato per l'elaborazione OLAP e le applicazioni di Business Intelligence (BI), che richiedono query complesse a fronte di grandi set di dati. Dato che soddisfa requisiti molto diversi, lo schema di archiviazione dati specializzato e il motore di esecuzione della query usati da Amazon Redshift sono completamente diversi dall'implementazione PostgreSQL. Ad esempio, se le applicazioni di elaborazione di transazioni online (OLTP) archiviano di solito i dati in righe, Amazon Redshift archivia i dati in colonne, usando codifiche di compressione dei dati specializzate per un utilizzo ottimale della memoria e dell'I/O su disco. Gli indici secondari e le operazioni di manipolazione dei dati a riga singola sono stati omessi per migliorare le prestazioni.

Consultare Panoramica del sistema e dell'architettura per una spiegazione dettagliata dell'architettura dei sistemi di data warehouse di Amazon Redshift.

PostgreSQL 9.x comprende alcune funzionalità che non sono supportate in Amazon Redshift. Inoltre, vi sono importanti differenze tra l'SQL di Amazon Redshift e PostgreSQL che è necessario tenere in considerazione: In questa sezione sono descritte le le differenze tra Amazon Redshift e PostgreSQL e sono fornite indicazioni per lo sviluppo di un data warehouse che sfrutti l'intera implementazione SQL di Amazon Redshift.

Amazon Redshift e JDBC e ODBC PostgreSQL

Poiché Amazon Redshift si basa su PostgreSQL, in precedenza abbiamo consigliato di usare il driver JDBC4 Postgresql versione 8.4.703 e i driver psqlODBC versione 9.x. Se attualmente sono utilizzati tali driver, consigliamo di passare ai nuovi driver specifici di Amazon Redshift. Per ulteriori informazioni sui driver e sulla configurazione delle connessioni, consulta Driver JDBC e ODBC per Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Per evitare out-of-memory errori sul lato client durante il recupero di set di dati di grandi dimensioni utilizzando JDBC, puoi consentire al client di recuperare i dati in batch impostando il parametro JDBC fetch size. Per ulteriori informazioni, consulta Impostazione del parametro delle dimensioni del recupero JDBC.

Amazon Redshift non riconosce il parametro JDBC maxRows. Specifica invece una clausola LIMIT per restringere il set di risultati. È anche possibile usare una OFFSET clausola per saltare a un punto di partenza specifico nel set di risultati.

Caratteristiche implementate in modo diverso

Molti elementi della sintassi SQL di Amazon Redshift hanno diverse caratteristiche a livello di prestazioni e usano sintassi e semantica piuttosto differenti dall'implementazione PostgreSQL equivalente.

> **Important**
>
> Non assumere che la sintassi degli elementi in comune tra Amazon Redshift e PostgreSQL sia identica. Assicurarsi di consultare Comandi SQL nella Guida per gli sviluppatori di database di Amazon Redshift per comprendere le differenze spesso minime.

Un esempio in particolare è il comando VACUUM, usato per pulire e riorganizzare le tabelle. VACUUM funziona diversamente e usa un set di parametri differente rispetto alla versione

PostgreSQL. Per ulteriori informazioni sull'uso di VACUUM con Amazon Redshift, consultare Vacuum delle tabelle.

Spesso, anche le caratteristiche e gli strumenti di amministrazione e gestione dei database sono diversi. Ad esempio, Amazon Redshift mantiene un set di visualizzazioni e tabelle di sistema che forniscono informazioni sulla modalità di funzionamento del sistema. Per ulteriori informazioni, consultare Tabelle e viste di sistema.

L'elenco seguente comprende alcuni esempi di funzionalità SQL implementate diversamente in Amazon Redshift.

- CREATE TABLE

  Amazon Redshift non supporta gli spazi tabelle, il partizionamento di tabella, l'ereditarietà e alcune limitazioni. L'implementazione di Amazon Redshift di CREATE TABLE consente di definire l'ordinamento e la distribuzione di algoritmi per tabelle in modo da ottimizzare l'elaborazione parallela.

  Amazon Redshift Spectrum supporta il partizionamento di tabella usando il comando CREATE EXTERNAL TABLE.
- ALTER TABLE

  È supportato solo un subset di operazioni ALTER COLUMN.

  ADD COLUMN supporta l'aggiunta di una sola colonna in ogni istruzione ALTER TABLE.
- COPY

  Il comando COPY di Amazon Redshift è altamente specializzato per consentire il caricamento di dati da bucket Amazon S3 e da tabelle Amazon DynamoDB e per facilitare la compressione automatica. Per informazioni dettagliate, consultare la sezione Caricamento dei dati e il riferimento al comando COPY.
- VACUUM

  I parametri per VACUUM sono completamente diversi. Ad esempio, l'operazione VACUUM di default in PostgreSQL reclama semplicemente spazio e lo rende disponibile per il riutilizzo. Tuttavia, l'operazione VACUUM di default in Amazon Redshift è VACUUM FULL, che rivendica spazio su disco e riordina tutte le righe.
- Gli spazi finali nei valori VARCHAR vengono ignorati quando i valori stringa vengono confrontati. Per ulteriori informazioni, consultare Significato degli spazi finali.

Caratteristiche PostgreSQL non supportate

Queste caratteristiche di PostgreSQL non sono supportate in Amazon Redshift.

> **Important**
>
> Non assumere che la sintassi degli elementi in comune tra Amazon Redshift e PostgreSQL sia identica. Assicurarsi di consultare Comandi SQL nella Guida per gli sviluppatori di database di Amazon Redshift per comprendere le differenze spesso minime.

- Lo strumento di query psql non è supportato. Il client Amazon Redshift RSQL è supportato.
- Partizionamento di tabella (partizionamento elenco e intervallo)
- Spazi tabelle
- Vincoli
  - Unique
  - Chiave esterna
  - Chiave primaria
  - Vincoli check
  - Vincoli di esclusione

  I vincoli di chiave esterna, chiave primaria e univoci sono consentiti, ma solo a scopo informativo. Non vengono applicati dal sistema, ma vengono utilizzati dal pianificatore della query.
- Ruoli database
- Ereditarietà
- Colonne di sistema PostgreSQL

  SQL di Amazon Redshift non definisce in modo implicito le colonne di sistema. Tuttavia, non è possibile usare i nomi delle colonne di sistema PostgreSQL come nomi di colonne definite dall'utente: `oid`, `tableoid`, `xmin`, `cmin`, `xmax`, `cmax` e `ctid`. Per ulteriori informazioni. consulta https://www.postgresql.org/docs/8.0/static/ddl-system-columns.html.
- Indici
- Clausola NULLS nelle funzioni finestra
- Regole di confronto

Amazon Redshift non supporta sequenze di regole di confronto definite dall'utente o specifiche in termini di impostazioni locali. Per informazioni, consultare Sequenze di regole di confronto.

- Espressioni valore
  - Espressioni con pedice
  - Costruttori di array
  - Costruttori di riga
- Trigger
- Gestione di dati esterni (SQL/MED)
- Funzioni di tabella
- Elenco VALUES usato come tabelle costanti
- Sequenze
- Ricerca full-text

Tipi di dati PostgreSQL non supportati

Solitamente, se una query tenta di usare un tipo di dati non supportato, compresi interpreti impliciti o espliciti, restituirà un errore. Tuttavia, alcune query che utilizzano tipi di dati non supportati saranno eseguite sul nodo principale ma non sui nodi di calcolo. Per informazioni, consultare Funzioni SQL supportate sul nodo principale.

Per un elenco dei tipi di dati supportati, consultare Tipi di dati.

Questi tipi di dati PostgreSQL non sono supportati in Amazon Redshift.

- Matrici
- BIT, BIT VARYING
- BYTEA
- Tipi compositi
- Tipi data/ora
- Tipi enumerati
- Tipi geometrici
- HSTORE
- JSON

- Tipi di indirizzo di rete
- Tipi numerici
  - SERIAL, BIGSERIAL, SMALLSERIAL
  - MONEY
- Tipi Identificatore di oggetto
- Pseudotipi
- Tipi di intervallo
- Tipi di caratteri speciali
  - "char": un tipo interno a singolo byte (in cui il tipo di dato denominato char è racchiuso tra virgolette).
  - name: un tipo interno per i nomi di oggetti.

  Per ulteriori informazioni su questi tipi, consultare Tipi di caratteri speciali nella documentazione PostgreSQL.
- Tipi di ricerca testo
- TXID_SNAPSHOT
- UUID
- XML

Funzioni PostgreSQL non supportate

Molte funzioni non escluse hanno un uso o una semantica diversa. Ad esempio, alcune funzioni supportate saranno eseguite solo sul nodo principale. Inoltre, alcune funzioni non supportate non restituiranno un errore se eseguite sul nodo principale. Il fatto che in alcuni casi queste funzioni non restituiscono un errore non significa che la funzione sia supportata da Amazon Redshift.

> **Important**
>
> Non assumere che la sintassi degli elementi in comune tra Amazon Redshift e PostgreSQL sia identica. Assicurarsi di consultare Comandi SQL nella Guida per gli sviluppatori di database di Amazon Redshift per comprendere le differenze spesso minime.

Per ulteriori informazioni, consultare Funzioni SQL supportate sul nodo principale.

Queste funzioni PostgreSQL non sono supportate in Amazon Redshift.

- Funzioni di richiesta di privilegi di accesso
- Funzioni di blocco consulenza
- Funzioni di aggregazione
  - STRING_AGG()
  - ARRAY_AGG()
  - EVERY()
  - XML_AGG()
  - CORR()
  - COVAR_POP()
  - COVAR_SAMP()
  - REGR_AVGX(), REGR_AVGY()
  - REGR_COUNT()
  - REGR_INTERCEPT()
  - REGR_R2()
  - REGR_SLOPE()
  - REGR_SXX(), REGR_SXY(), REGR_SYY()
- Operatori e funzioni della matrice
- Funzioni di controllo del backup
- Funzioni di informazione sui commenti
- Funzioni sulla posizione dell'oggetto Database
- Funzioni sulle dimensioni dell'oggetto Database
- Operatori e funzioni data e ora
  - CLOCK_TIMESTAMP()
  - JUSTIFY_DAYS(), JUSTIFY_HOURS(), JUSTIFY_INTERVAL()
  - PG_SLEEP()
  - TRANSACTION_TIMESTAMP()
- Funzioni di supporto ENUM
- Operatori e funzioni geometriche
- Funzioni di accesso a file generiche

- IS DISTINCT FROM
- Operatori e funzioni sull'indirizzo di rete
- Funzioni matematiche
  - DIV()
  - SETSEED()
  - WIDTH_BUCKET()
- Funzioni di restituzione di set
  - GENERATE_SERIES()
  - GENERATE_SUBSCRIPTS()
- Operatori e funzioni di intervallo
- Funzioni di controllo del ripristino
- Funzioni di informazione sul ripristino
- Funzione ROLLBACK TO SAVEPOINT
- Funzioni di richiesta di visibilità dello schema
- Funzioni di segnalazione server
- Funzioni di sincronizzazione di snapshot
- Funzioni di manipolazione di sequenze
- Funzioni stringa
  - BIT_LENGTH()
  - OVERLAY()
  - CONVERT(), CONVERT_FROM(), CONVERT_TO()
  - ENCODE()
  - FORMAT()
  - QUOTE_NULLABLE()
  - REGEXP_MATCHES()
  - REGEXP_SPLIT_TO_ARRAY()
  - REGEXP_SPLIT_TO_TABLE()
- Funzioni di informazione su catalogo di sistema
- Funzioni di informazioni di sistema
  - CURRENT_CATALOG CURRENT_QUERY()

- INET_CLIENT_ADDR()
- INET_CLIENT_PORT()
- INET_SERVER_ADDR() INET_SERVER_PORT()
- PG_CONF_LOAD_TIME()
- PG_IS_OTHER_TEMP_SCHEMA()
- PG_LISTENING_CHANNELS()
- PG_MY_TEMP_SCHEMA()
- PG_POSTMASTER_START_TIME()
- PG_TRIGGER_DEPTH()
- SHOW VERSION()

- Funzioni e operatori di ricerca di testo
- Funzioni di snapshot e ID transazioni
- Funzioni di trigger
- Funzioni XML

Uso di SQL

Argomenti

Il linguaggio SQL è composto da comandi e funzioni che usi per lavorare con database e oggetti di database. Il linguaggio applica anche regole relative all'uso di tipi di dati, espressioni e letterali.

Convenzioni del riferimento SQL

Questa sezione descrive le convenzioni utilizzate per scrivere la sintassi per le funzioni, i comandi e le espressioni SQL descritte nella sezione del riferimento SQL.

Carattere	Descrizione
CAPS	Le parole in lettere maiuscole sono parole chiave.
[ ]	Le parentesi indicano argomenti opzionali. Più argomenti tra parentesi indicano che è possibile scegliere qualsiasi numero degli argomenti. Inoltre, gli argomenti tra parentesi su righe separate indicano che il parser di Amazon Redshift prevede che gli argomenti siano nell'ordine in cui sono elencati nella sintassi. Per un esempio, consultare SELECT.
{ }	Le parentesi graffe indicano che è necessario scegliere uno degli argomenti racchiusi nelle stesse.
|	Le pipe indicano che è possibile scegliere tra gli argomenti.
corsivo	Le parole in corsivo indicano dei segnaposto. Devi inserire il valore appropria to al posto della parola in corsivo.
. . .	I puntini di sospensione indicano che è possibile ripetere l'elemento precedente.
'	Le parole tra virgolette singole indicano che è necessario digitare le virgolett e.

Elementi base

Argomenti

- Nomi e identificatori
- Valori letterali
- Nulls
- Tipi di dati
- Sequenze di regole di confronto

Questa sezione comprende le regole per lavorare con nomi oggetto di database, valori letterali, null e tipi di dati.

Nomi e identificatori

I nomi identificano oggetti di database, comprese tabelle e colonne, nonché utenti e password. È possibile usare i termini nome e identificatore in modo intercambiabile. Ci sono due tipi di identificatori, gli identificatori standard e gli identificatori delimitati o racchiusi tra virgolette. È necessario che gli identificatori contengano solo caratteri stampabili UTF-8. Le lettere ASCII negli identificatori standard e delimitati fanno distinzione tra maiuscole e minuscole e sono scritte in minuscolo nel database. Nei risultati delle query, i nomi delle colonne vengono restituiti in lettere minuscole per impostazione predefinita. Per restituire i nomi delle colonne in lettere maiuscole, impostare il parametro di configurazione describe_field_name_in_uppercase su **`true`**.

Identificatori standard

Gli identificatori SQL standard rispettano un insieme di regole ed è necessario che:

- Inizino con un carattere alfabetico o di sottolineatura a byte singolo ASCII oppure con un carattere multibyte UTF-8 da due a quattro byte di lunghezza.
- È possibile che i caratteri seguenti siano caratteri alfanumerici, di sottolineatura o segni di dollaro a byte singolo ASCII oppure caratteri multibyte UTF-8 da due a quattro byte di lunghezza.
- Abbiano una lunghezza compresa tra 1 e 127 byte, senza comprendere le virgolette per gli identificatori delimitati.
- Non contengano virgolette e spazi.
- Non siano una parola chiave di SQL riservata.

Identificatori delimitati

Gli identificatori delimitati (conosciuti anche come identificatori tra virgolette) iniziano e terminano con le virgolette doppie ("). Se vedi un identificatore delimitato, è necessario usare le virgolette doppie per ogni riferimento a quell'oggetto. L'identificatore può contenere qualsiasi carattere stampabile UTF-8 standard diverso dalle virgolette doppie. Pertanto, è possibile creare nomi di colonna o tabella che comprendono caratteri altrimenti non ammessi, come spazi o il simbolo di percentuale.

Le lettere ASCII negli identificatori delimitati fanno distinzione tra maiuscole e minuscole e sono scritte in minuscolo. Per usare una virgoletta doppia in una stringa, è necessario farla precedere da un altro carattere di virgoletta doppia.

Identificatori con distinzione maiuscolo/minuscolo

Gli identificatori con distinzione tra maiuscole e minuscole (noti anche come identificatori misti) possono contenere lettere maiuscole e minuscole. Per utilizzare gli identificatori con distinzione tra maiuscole e minuscole, è possibile impostare la configurazione da `enable_case_sensitive_identifier` a `true`. È possibile impostare questa configurazione per il cluster o per una sessione. Per ulteriori informazioni, consulta Valori di parametro predefiniti nella Guida alla gestione di Amazon Redshift e enable_case_sensitive_identifier.

Nomi di colonna di sistema

Non è possibile usare i seguenti i nomi delle colonne di sistema PostgreSQL come nomi delle colonne definite dall'utente. Per ulteriori informazioni. consulta https://www.postgresql.org/docs/8.0/static/ddl-system-columns.html.

- `oid`
- `tableoid`
- `xmin`
- `cmin`
- `xmax`
- `cmax`
- `ctid`

Esempi

Questa tabella mostra esempi di identificatori delimitati, l'output risultante e una discussione:

Sintassi	Risultato	Discussione
"group"	gruppo	GROUP è una parola riservata, quindi il suo uso all'interno di un identificatore necessita delle virgolette doppie.
"""WHERE"""	"dove"	Anche WHERE è una parola riservata. Per includere le virgolette nella stringa, eseguire l'escape per ciascuna virgoletta utilizzando virgolette supplementari.

Sintassi	Risultato	Discussione
"This name"	this name	Le doppie virgolette sono necessarie per conservare lo spazio.
"This ""IS IT"""	this "is it"	Perché diventino parte del nome, è necessario che ciascuna delle virgolette che circonda IS IT sia preceduta da altre virgolette.

Per creare una tabella chiamata group con una colonna di nome this "is it":

```
create table "group" (
"This ""IS IT""" char(10));
```

Le seguenti query restituiscono lo stesso risultato:

```
select "This ""IS IT"""
from "group";

this "is it"
--------------
(0 rows)
```

```
select "this ""is it"""
from "group";

this "is it"
--------------
(0 rows)
```

Anche la seguente sintassi `table.column` completa restituisce lo stesso risultato:

```
select "group"."this ""is it"""
from "group";

this "is it"
--------------
(0 rows)
```

Il seguente comando CREATE TABLE crea una tabella con una barra nel nome di una colonna:

```
create table if not exists city_slash_id(
                  "city/id" integer not null,
                  state char(2) not null);
```

Valori letterali

Un valore letterale o una costante è un valore di dati fisso, composto da una sequenza di caratteri o da una costante numerica. Amazon Redshift supporta diversi tipi di letterali, tra cui:

- Valori letterali per numeri interi, decimali e in virgola mobile. Per ulteriori informazioni, consultare Valori letterali interi e in virgola mobile.
- Valori letterali carattere, definiti anche come stringe, stringhe di caratteri o costanti di caratteri
- Valori letterali di intervallo e datetime, usati con i tipi di dati datetime. Per ulteriori informazioni, consultare Valori letterali di data, ora e timestamp e Tipi di dati e valori letterali a intervalli.

Nulls

Se una colonna in una riga è mancante, sconosciuta o non applicabile, è un valore null o viene detta contenere un valore null. I valori null possono comparire nei campi di qualsiasi tipo di dati non limitati da una chiave primaria o da vincoli NOT NULL. Un valore null non equivale al valore zero o a una stringa vuota.

Qualsiasi espressione aritmetica contenente un valore null viene sempre valutata come valore null. Tutti gli operatori, ad eccezione della concatenazione, restituiscono un valore null quando gli viene fornito un operando o un argomento null.

Per testare valori null, usa le condizioni di confronto IS NULL e IS NOT NULL. Dato che un valore null rappresenta una mancanza di dati, un valore null non è uguale o disuguale a nessun valore o a un altro valore null.

Tipi di dati

Argomenti

- Caratteri multibyte
- Tipi numerici
- Tipi di carattere

- Tipi datetime
- Tipo booleano
- Tipo HLLSKETCH
- Tipo SUPER
- Tipo VARBYTE
- Conversione e compatibilità dei tipi

Ciascun valore che Amazon Redshift memorizza o recupera ha un tipo di dati con un set fisso di proprietà associate. I tipi di dati vengono dichiarati al momento della creazione delle tabelle. Un tipo di dati limita il set di valori che un argomento o una colonna può contenere.

Nella tabella seguente sono elencati i tipi di dati che è possibile usare nelle tabelle di Amazon Redshift.

Tipo di dati	Alias	Descrizione
SMALLINT	INT2	Intero a due byte firmato
INTEGER	INT, INT4	Intero a quattro byte firmato
BIGINT	INT8	Intero a otto byte firmato
DECIMAL	NUMERIC	Numerico esatto di precisione selezionabile
REAL	FLOAT4	Numero in virgola mobile a precisione singola
DOUBLE PRECISION	FLOAT8, FLOAT	Numero in virgola mobile a precisione doppia
CHAR	CHARACTER, NCHAR, BPCHAR	Stringa di caratteri a lunghezza fissa
VARCHAR	CHARACTER VARYING, NVARCHAR, TEXT	Stringa di caratteri a lunghezza variabile con un limite definito dall'utente

Tipo di dati	Alias	Descrizione
DATE		Data di calendario (anno, mese, giorno)
TIME	TIME WITHOUT TIME ZONE	Ora del giorno
TIMETZ	TIME WITH TIME ZONE	Ora del giorno con fuso orario
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE	Data e ora (senza fuso orario)
TIMESTAMPTZ	TIMESTAMP WITH TIME ZONE	Data e ora (con fuso orario)
INTERVAL YEAR TO MONTH		Durata del periodo in ordine anno-mese
INTERVAL DAY TO SECOND		Durata del tempo espressa in base al secondo ordine
BOOLEAN	BOOL	Booleani logici (true/false)
HLLSKETCH		Tipo usato con gli HyperLogL og schizzi.
SUPER		Un tipo di dati superset che comprende tutti i tipi scalari di Amazon Redshift, inclusi i tipi complessi come ARRAY e STRUCTS.
VARBYTE	VARBINARY, BINARY VARYING	Valore binario a lunghezza variabile
GEOMETRY		Dati spaziali
GEOGRAPHY		Dati spaziali

> **Note**
>
> Per informazioni sui tipi di dati non supportati, come “char” (da notare che char è racchiuso tra virgolette), consultare Tipi di dati PostgreSQL non supportati.

Caratteri multibyte

Il tipo di dati VARCHAR supporta caratteri multibyte UTF-8 fino a un massimo di quattro byte. I caratteri a cinque byte o più non sono supportati. Per calcolare la dimensione di una colonna VARCHAR che contiene caratteri multibyte, moltiplica il numero di caratteri per il numero di byte per carattere. Ad esempio, se una stringa ha quattro caratteri cinesi e ciascun carattere è lungo tre byte, allora avrai bisogno di una colonna VARCHAR(12) per memorizzare la stringa.

Il tipo di dati VARCHAR non supporta i punti di codice UTF-8 non validi seguenti:

`0xD800 – 0xDFFF` (Sequenze di byte: `ED A0 80` – `ED BF BF`)

Il tipo di dati CHAR non supporta caratteri multibyte.

Tipi numerici

Argomenti

- Tipi Integer
- Tipo DECIMAL o NUMERIC
- Note sull'utilizzo di colonne NUMERIC o DECIMAL a 128 bit
- Tipi in virgola mobile
- Calcoli con valori numerici
- Valori letterali interi e in virgola mobile
- Esempi con tipi numerici

I tipi di dati numerici comprendono numeri interi, decimali e in virgola mobile.

Tipi Integer

Usa i tipi di dati SMALLINT, INTEGER e BIGINT per memorizzare numeri interi di intervalli diversi. Non è possibile salvare valori non compresi nell'intervallo consentito per ciascun tipo.

Nome	Storage	Intervallo
SMALLINT o INT2	2 byte	Da -32768 a +32767
INTEGER, INT o INT4	4 byte	Da -2147483648 a +2147483647
BIGINT o INT8	8 byte	Da -92233720 36854775808 a 922337203 6854775807

Tipo DECIMAL o NUMERIC

Usare il tipo di dati DECIMAL o NUMERIC per memorizzare i valori con una precisione definita dall'utente. Le parole chiave DECIMAL e NUMERIC sono interscambiabili. In questo documento, decimale è il termine preferito per questo tipo di dati. Il termine numerici è usato solitamente per riferirsi a tipi di dati interi, decimali e in virgola mobile.

Storage	Intervallo
Variabile, fino a 128 bit per i tipi DECIMAL non compressi.	Interi firmati da 128 bit con fino a 38 cifre di precisione.

Definisci una colonna DECIMAL in una tabella specificando precisione e scala:

```
decimal(precision, scale)
```

precisione

Il numero totale di cifre significative nell'intero valore: il numero di cifre su entrambi i lati del punto decimale. Ad esempio, il numero `48.2891` ha una precisione di 6 e una scala di 4. La precisione predefinita, se non specificata, è 18. La precisione massima è 38.

Se il numero di cifre alla sinistra del punto decimale in un valore input supera la precisione della colonna meno la sua scala, non è possibile copiare il valore nella colonna (o inserito

o aggiornato). Questa regola si applica a qualsiasi valore che non rientra nell'intervallo della definizione della colonna. Ad esempio, gli intervalli di valori ammessi per una colonna `numeric(5,2)` è da `-999.99` a `999.99`.

scale

Il numero di cifre decimale nella parte frazionaria del valore, alla destra del punto decimale. Gli interi hanno una scala di zero. Nella specifica di una colonna, è necessario che il valore della scala sia inferiore o uguale al valore della precisione. La scala predefinita, se non specificata, è 0. La scala massima è 37.

Se la scala di un valore input caricato in una tabella è maggiore della scala della colonna, il valore viene arrotondato alla scala specificata. Ad esempio, la colonna PRICEPAID nella tabella SALES è una colonna DECIMAL(8,2). Se un valore DECIMAL(8,4) viene inserito nella colonna PRICEPAID, il valore viene arrotondato alla scala di 2.

```
insert into sales
values (0, 8, 1, 1, 2000, 14, 5, 4323.8951, 11.00, null);

select pricepaid, salesid from sales where salesid=0;

pricepaid | salesid
-----------+---------
4323.90 |       0
(1 row)
```

Tuttavia, i risultati di espliciti cast di valori selezionati dalle tabelle non sono arrotondati.

> **Note**
>
> Il valore positivo massimo che è possibile inserire in una colonna DECIMAL(19,0) è `9223372036854775807` (2^{63} -1). Il valore negativo massimo è `-9223372036854775807`. Ad esempio, il tentativo di inserire il valore `9999999999999999` (19 nove) causerà un errore dell'overflow. Indipendentemente dalla posizione del punto decimale, la stringa più grande che Amazon Redshift può rappresentare come numero DECIMAL è `9223372036854775807`. Ad esempio, il valore più grande che è possibile caricare in una colonna DECIMAL(19,18) è `9.223372036854775807`.
> Queste regole sono dovute al fatto che i valori di tipo DECIMAL con 19 o meno cifre di precisione significative vengono archiviati internamente come numeri interi a 8 byte, mentre

i valori di tipo DECIMAL con 20-38 cifre di precisione significative vengono archiviati come numeri interi a 16 byte.

Note sull'utilizzo di colonne NUMERIC o DECIMAL a 128 bit

Non assegnare in modo arbitrario la precisione massima alle colonne DECIMAL a meno che non sia certo che l'applicazione richieda tale precisione. I valori a 128 bit usano il doppio dello spazio su disco rispetto ai valori a 64 bit e possono quindi rallentare il tempo di esecuzione delle query.

Tipi in virgola mobile

Usa i tipi di dati REAL e DOUBLE PRECISION per memorizzare valori numerici con precisione variabile. Questi tipi sono inesatti, il che significa che alcuni valori vengono memorizzati come approssimazioni, così che la memorizzazione e la restituzione di un valore specifico possono risultare in lievi discrepanze. Se hai bisogno di calcoli e storage precisi (come per importi monetari), usa il tipo di dati DECIMAL.

REAL rappresenta il formato a virgola mobile a precisione singola, secondo lo standard IEEE 754 per l'aritmetica binaria a virgola mobile. Ha una precisione di circa 6 cifre e un intervallo compreso tra 1E-37 e 1E+37. È inoltre possibile specificare questo tipo di dati come FLOAT4.

DOUBLE PRECISION rappresenta il formato a virgola mobile a precisione doppia, secondo lo standard IEEE 754 per l'aritmetica binaria a virgola mobile. Ha una precisione di circa 15 cifre e un intervallo compreso tra 1E-307 e 1E+308. È inoltre possibile specificare questo tipo di dati come FLOAT o FLOAT8.

Oltre ai valori numerici ordinari, i tipi a virgola mobile hanno diversi valori speciali. Usa le virgolette singole per racchiudere questi valori quando li usi in un'istruzione SQL:

- `NaN` – not-a-number
- `Infinity`: infinito
- `-Infinity`: infinito negativo

Ad esempio, per inserire not-a-number nella colonna `day_charge` della tabella `customer_activity` esegui il seguente codice SQL:

```
insert into customer_activity(day_charge) values('NaN');
```

Calcoli con valori numerici

In questo contesto, calcolo si riferisce a operazioni matematiche binarie: addizione, sottrazione, moltiplicazione e divisione. Questa sezione descrive i tipi restituiti previsti per queste operazioni, nonché la formula specifica che viene applicata per determinare la precisione e la scala quando sono coinvolti tipi di dati DECIMAL.

Quando valori numerici vengono calcolati durante l'elaborazione di query, potresti affrontare casi in cui il calcolo è impossibile e la query restituisce un errore dell'overflow numerico. Potresti anche riscontrare casi in cui la scala di valori calcolati varia o è imprevista. Per alcune operazioni, per risolvere questi problemi, è possibile usare il casting esplicito (promozione del tipo) o i parametri di configurazione di Amazon Redshift.

Per informazioni sui risultati di calcoli simili con funzioni SQL, consultare Funzioni di aggregazione.

Tipi restituiti per i calcoli

Dato il set di tipi di dati numerici supportato in Amazon Redshift, la tabella seguente mostra i tipi restituiti previsti per operazioni di addizione, sottrazione, moltiplicazione e divisione. La prima colonna sul lato sinistro della tabella rappresenta il primo operando nel calcolo e la riga in alto rappresenta il secondo operando.

	INT2	INT4	INT8	DECIMAL	FLOAT4	FLOAT8
INT2	INT2	INT4	INT8	DECIMAL	FLOAT8	FLOAT8
INT4	INT4	INT4	INT8	DECIMAL	FLOAT8	FLOAT8
INT8	INT8	INT8	INT8	DECIMAL	FLOAT8	FLOAT8
DECIMAL	DECIMAL	DECIMAL	DECIMAL	DECIMAL	FLOAT8	FLOAT8
FLOAT4	FLOAT8	FLOAT8	FLOAT8	FLOAT8	FLOAT4	FLOAT8
FLOAT8	FLOAT8	FLOAT8	FLOAT8	FLOAT8	FLOAT8	FLOAT8

Precisione e scala di risultati DECIMAL calcolati

La tabella seguente riassume le regole per la scala e la precisione risultanti dal calcolo quando operazioni matematiche restituiscono risultati DECIMAL. In questa tabella, `p1` e `s1` rappresentano

la precisione e la scala del primo operando in un calcolo e p2 e s2 rappresentano la precisione e la scala del secondo operando. (Indipendentemente da questi calcoli, la precisione del risultato massima è 38 e la scala del risultato massima è 38.)

Operazione	Scala e precisione del risultato
+ oppure -	Dimensionare = `max(s1,s2)` Precisione = `max(p1-s1,p2-s2)+1+scale`
*	Dimensionare = `s1+s2` Precisione = `p1+p2+1`
/	Dimensionare = `max(4,s1+p2-s2+1)` Precisione = `p1-s1+ s2+scale`

Ad esempio, le colonne PRICEPAID e COMMISSION nella tabella SALES sono entrambe colonne DECIMAL(8,2). Se dividi PRICEPAID per COMMISSION (o viceversa), la formula è applicata come segue:

```
Precision = 8-2 + 2 + max(4,2+8-2+1)
= 6 + 2 + 9 = 17

Scale = max(4,2+8-2+1) = 9

Result = DECIMAL(17,9)
```

Il calcolo seguente è la regola generale per il calcolo della scala e della precisione risultanti per le operazioni eseguite su valori DECIMAL con operatori impostati come UNION, INTERSECT ed EXCEPT o funzioni come COALESCE e DECODE:

```
Scale = max(s1,s2)
Precision = min(max(p1-s1,p2-s2)+scale,19)
```

Ad esempio, una tabella DEC1 con una colonna DECIMAL(7,2) viene unita con una tabella DEC2 con una colonna DECIMAL(15,3) per creare una tabella DEC3. Lo schema di DEC3 mostra che la colonna diventa una colonna NUMERIC(15,3).

```
create table dec3 as select * from dec1 union select * from dec2;
```

Risultato

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'dec3';

column |     type      | encoding | distkey | sortkey
-------+---------------+----------+---------+---------
c1     | numeric(15,3) | none     | f       | 0
```

Nell'esempio sopra, la formula è applicata come segue:

```
Precision = min(max(7-2,15-3) + max(2,3), 19)
= 12 + 3 = 15

Scale = max(2,3) = 3

Result = DECIMAL(15,3)
```

Note sulle operazioni di divisione

Per le operazioni di divisione, divide-by-zero le condizioni restituiscono errori.

Il limite di scala di 100 è applicato dopo aver calcolato la precisione e la scala. Se la scala del risultato calcolata è maggiore di 100, i risultati della divisione vengono scalati come segue:

- Precisione = `precision - (scale - max_scale)`
- Dimensionare = `max_scale`

Se la precisione calcolata supera la precisione massima (38), la precisione viene ridotta a 38 e la scala diventa il risultato di: `max((38 + scale - precision), min(4, 100))`

Condizioni di overflow

L'overflow viene controllato per tutti i calcoli numerici. I dati DECIMAL con una precisione di 19 o inferiore vengono memorizzati come interi a 64 bit. I dati DECIMAL con una precisione superiore a 19 vengono memorizzati come interi a 128 bit. La precisione massima per tutti i valori DECIMAL è 38 e

la scala massima è 37. Gli errori dell'overflow si verificano quando un valore supera questi limiti, che si applicano agli insiemi dei risultati finali e intermedi:

- Il casting esplicito risulta in errori dell'overflow runtime quando valori di dati specifici non rientrano nella scala o nella precisione richiesta specificata dalla funzione cast. Ad esempio, non è possibile eseguire il cast di tutti i valori dalla colonna PRICEPAID nella tabella SALES (una colonna DECIMAL(8,2)) e restituire un risultato DECIMAL(7,3):

  ```
  select pricepaid::decimal(7,3) from sales;
  ERROR:  Numeric data overflow (result precision)
  ```

  Questo errore si verifica perché non è possibile eseguire il cast di alcuni dei valori più grandi nella colonna PRICEPAID.

- Le operazioni di moltiplicazione producono risultati in cui la scala del risultato è la somma della scala di ciascun operando. Se entrambi gli operando hanno una scala di 4, ad esempio, la scala del risultato è 8, lasciando solo 10 cifre per il lato sinistro del punto decimale. Pertanto, è relativamente facile incorrere in condizioni di overflow quando si moltiplicano due grandi numeri che possiedono entrambi una scala significativa.

  Il seguente codice restituisce un errore di overflow.

  ```
  SELECT CAST(1 AS DECIMAL(38, 20)) * CAST(10 AS DECIMAL(38, 20));
  ERROR: 128 bit numeric data overflow (multiplication)
  ```

  Puoi risolvere l'errore di overflow utilizzando la divisione anziché la moltiplicazione. Utilizza l'esempio seguente per dividere per 1 diviso per il divisore originale.

  ```
  SELECT CAST(1 AS DECIMAL(38, 20)) / (1 / CAST(10 AS DECIMAL(38, 20)));
  +----------+
  | ?column? |
  +----------+
  | 10       |
  +----------+
  ```

Calcoli numerici con tipi INTEGER e DECIMAL

Quando uno degli operandi in un calcolo ha un tipo di dati INTEGER e l'altro operando è DECIMAL, viene implicitamente eseguito il cast dell'operando INTEGER a DECIMAL:

- Per INT2 (SMALLINT) viene eseguito il cast a DECIMAL(5,0)
- Per INT4 (INTEGER) viene eseguito il cast a DECIMAL(10,0)
- Per INT8 (BIGINT) viene eseguito il cast a DECIMAL(19,0)

Ad esempio, se moltiplichi SALES.COMMISSION, una colonna DECIMAL(8,2), e SALES.QTYSOLD, una colonna SMALLINT, per questo calcolo viene eseguito il cast come segue:

```
DECIMAL(8,2) * DECIMAL(5,0)
```

Valori letterali interi e in virgola mobile

I valori letterali o le costanti che rappresentano numeri possono essere interi o in virgola mobile.

Valori letterali interi

Una costante intera è una sequenza delle cifre 0-9, con un segno positivo (+) o negativo (-) opzionale che le precede.

Sintassi

```
[ + | - ] digit ...
```

Esempi

I valori letterali interi includono i seguenti:

```
23
-555
+17
```

Valori letterali in virgola mobile

I valori letterali in virgola mobile (definiti anche valori letterali decimali, numerici o frazionali) sono sequenze di cifre che possono comprendere un punto decimale e, opzionalmente, il segno dell'esponente (e).

Sintassi

```
[ + | - ] digit ... [ . ] [ digit ...]
```

```
[ e | E [ + | - ] digit ... ]
```

Argomenti

e | E

e o E indica che il numero è specificato in notazione scientifica.

Esempi

I valori letterali in virgola mobile validi includono i seguenti:

```
3.14159
-37.
2.0e19
-2E-19
```

Esempi con tipi numerici

Istruzione CREATE TABLE

La seguente istruzione CREATE TABLE dimostra la dichiarazione di diversi tipi di dati numerici:

```
create table film (
film_id integer,
language_id smallint,
original_language_id smallint,
rental_duration smallint default 3,
rental_rate numeric(4,2) default 4.99,
length smallint,
replacement_cost real default 25.00);
```

Tentativo di inserire un intero che non rientra nell'intervallo

L'esempio seguente tenta di inserire il valore 33.000 in una colonna SMALLINT.

```
insert into film(language_id) values(33000);
```

L'intervallo per SMALLINT è da -32.768 a +32.767, quindi Amazon Redshift restituisce un errore.

```
An error occurred when executing the SQL command:
insert into film(language_id) values(33000)

ERROR: smallint out of range [SQL State=22003]
```

Inserimento di un valore decimale in una colonna intera

L'esempio seguente inserisce il valore decimale in una colonna INT.

```
insert into film(language_id) values(1.5);
```

Questo valore viene inserito ma arrotondato al valore intero 2.

Inserimento di un decimale che ha esito positivo perché la sua scala viene arrotondata

L'esempio seguente inserisce un valore decimale che ha una precisione maggiore rispetto alla colonna.

```
insert into film(rental_rate) values(35.512);
```

In questo caso, il valore `35.51` viene inserito nella colonna.

Tentativo di inserire un valore decimale che non rientra nell'intervallo

In questo caso, il valore `350.10` non rientra nell'intervallo. Il numero di cifre per i valori nelle colonne DECIMAL è uguale alla precisione della colonna meno la sua scala (4 meno 2 per la colonna RENTAL_RATE). In altre parole, l'intervallo consentito per una colonna `DECIMAL(4,2)` va da `-99.99` a `99.99`.

```
insert into film(rental_rate) values (350.10);
ERROR:  numeric field overflow
DETAIL:  The absolute value is greater than or equal to 10^2 for field with precision
 4, scale 2.
```

Inserimento di valori a precisione variabile in una colonna REAL

L'esempio seguente inserisce valori a precisione variabile in una colonna REAL.

```
insert into film(replacement_cost) values(1999999.99);

insert into film(replacement_cost) values(1999.99);
```

```
select replacement_cost from film;

+------------------+
| replacement_cost |
+------------------+
| 2000000          |
| 1999.99          |
+------------------+
```

Il valore `1999999.99` viene convertito in `2000000` per soddisfare il requisito di precisione della colonna `REAL`. Il valore `1999.99` viene caricato così com'è.

Tipi di carattere

Argomenti

- Storage e intervalli
- CHAR o CHARACTER
- VARCHAR o CHARACTER VARYING
- Tipi NCHAR e NVARCHAR
- Tipi TEXT e BPCHAR
- Significato degli spazi finali
- Esempi con tipi di caratteri

I tipi di dati carattere comprendono CHAR (carattere) e VARCHAR (carattere variabile).

Storage e intervalli

I tipi di dati CHAR e VARCHAR sono definiti in termini di byte, non caratteri. Una colonna CHAR può contenere solo caratteri a byte singolo, quindi una colonna CHAR(10) può contenere una stringa con una lunghezza massima di 10 byte. Una VARCHAR può contenere caratteri multibyte fino a un massimo di quattro byte per carattere. Ad esempio, una colonna VARCHAR(12) può contenere 12 caratteri a byte singolo, 6 caratteri a due byte, 4 caratteri a tre byte o 3 caratteri a quattro byte.

Nome	Storage	Intervallo (larghezza della colonna)
CHAR, CHARACTER o NCHAR	Lunghezza della stringa, compresi	4096 byte

Nome	Storage	Intervallo (larghezza della colonna)
	spazi finali (se presenti)	
VARCHAR, CHARACTER VARYING o NVARCHAR	4 byte + byte totali per carattere, dove ogni carattere può essere da 1 a 4 byte.	65.535 bytes (64K -1)
BPCHAR	Conversione in CHAR(256) a lunghezza fissa.	256 byte
TEXT	Conversione in VARCHAR(256).	260 byte

> **Note**
>
> La sintassi CREATE TABLE supporta la parola chiave MAX per i tipi di dati carattere. Ad esempio:
>
> ```
> create table test(col1 varchar(max));
> ```
>
> L'impostazione MAX definisce la larghezza della colonna come 4.096 byte per CHAR o 65.535 byte per VARCHAR.

CHAR o CHARACTER

Usa una colonna CHAR o CHARACTER per memorizzare stringhe di lunghezza fissa. A queste stringhe vengono aggiunti spazi, quindi una colonna CHAR(10) occupa 10 byte di storage.

```
char(10)
```

Una colonna CHAR senza una specificazione di lunghezza risulta in una colonna CHAR(1).

VARCHAR o CHARACTER VARYING

Usa una colonna VARCHAR o CHARACTER VARYING per memorizzare stringhe di lunghezza variabile con un limite fisso. A queste stringhe non vengono aggiunti spazi vuoti, quindi una colonna VARCHAR(120) consiste di un massimo di 120 caratteri a byte singolo, 60 caratteri a due byte, 40 caratteri a tre byte o 30 caratteri a quattro byte.

```
varchar(120)
```

Se si utilizza il tipo di dati VARCHAR senza specificare la lunghezza in un'istruzione CREATE TABLE, la lunghezza predefinita è 256. Se utilizzata in un'espressione, la dimensione dell'output viene determinata utilizzando l'espressione di input (fino a 65535).

Tipi NCHAR e NVARCHAR

È possibile creare colonne con i tipi NCHAR e NVARCHAR (noti anche come tipi NATIONAL CHARACTER e NATIONAL CHARACTER VARYING). Questi tipi vengono convertiti in tipi CHAR e VARCHAR e vengono memorizzati nel numero di byte specificato.

Una colonna NCHAR senza una specificazione di lunghezza viene convertita in una colonna CHAR(1).

Una colonna NVARCHAR senza una specificazione di lunghezza viene convertita in una colonna VARCHAR(256).

Tipi TEXT e BPCHAR

È possibile creare una tabella Amazon Redshift con una colonna TEXT ma viene convertita in una colonna VARCHAR(256) che accetta valori di lunghezza variabile con un massimo di 256 caratteri.

È possibile creare una colonna Amazon Redshift con un tipo BPCHAR (caratteri con spazi aggiunti), che Amazon Redshift converte in una colonna CHAR(256) di lunghezza fissa.

Significato degli spazi finali

Entrambi i tipi di dati CHAR e VARCHAR memorizzano stringhe fino a n byte di lunghezza. Un tentativo di memorizzare una stringa più lunga in una colonna di questi tipi risulta in un errore, a meno che i caratteri extra non siano tutti spazi (vuoti); in questo caso la stringa viene troncata alla lunghezza massima. Se la stringa è più corta della lunghezza massima, ai valori CHAR vengono aggiunti spazi, ma i valori VARCHAR memorizzano la stringa senza spazi.

Gli spazi inziali nei valori CHAR sono sempre privi di significato dal punto di vista semantico. Vengono ignorati quando confronti due valori CHAR, non compresi nei calcoli LENGTH, e rimossi quando converti un valore CHAR in un altro tipo di stringa.

Gli spazi finali nei valori VARCHAR e CHAR vengono trattati come insignificanti dal punto di vista semantico quando i valori vengono confrontati.

I calcoli della lunghezza restituiscono la lunghezza delle stringhe di caratteri VARCHAR con spazi finali compresi nella lunghezza. Gli spazi finali non vengono contati nella lunghezza per le stringhe di caratteri a lunghezza fissa.

Esempi con tipi di caratteri

Istruzione CREATE TABLE

La seguente istruzione CREATE TABLE dimostra l'uso dei tipi di dati VARCHAR e CHAR:

```
create table address(
address_id integer,
address1 varchar(100),
address2 varchar(50),
district varchar(20),
city_name char(20),
state char(2),
postal_code char(5)
);
```

Gli esempi seguenti usano questa tabella.

Spazi finali in stringhe di caratteri a lunghezza variabile

Dato che ADDRESS1 è una colonna VARCHAR, gli spazi finali nel secondo indirizzo inserito sono insignificanti dal punto di vista semantico. In altre parole, questi due indirizzi inseriti corrispondono.

```
insert into address(address1) values('9516 Magnolia Boulevard');

insert into address(address1) values('9516 Magnolia Boulevard  ');
```

```
select count(*) from address
where address1='9516 Magnolia Boulevard';

count
```

```
-------
2
(1 row)
```

Se la colonna ADDRESS1 fosse una colonna CHAR e venissero inseriti gli stessi valori, la query COUNT(*) riconoscerebbe le stringhe di caratteri come uguali e restituirebbe 2.

Risultati della funzione LENGTH

La funzione LENGTH riconosce gli spazi finali nelle colonne VARCHAR:

```
select length(address1) from address;

length
--------
23
25
(2 rows)
```

Un valore di `Augusta` nella colonna CITY_NAME, che è una colonna CHAR, restituirebbe sempre una lunghezza di 7 caratteri, indipendentemente da qualsiasi spazio finale nella stringa input.

Valori che superano la lunghezza della colonna

Le stringhe di caratteri non vengono troncate per rientrare nella larghezza della colonna dichiarata:

```
insert into address(city_name) values('City of South San Francisco');
ERROR: value too long for type character(20)
```

Per ovviare a questo problema puoi eseguire il cast del valore alla dimensione della colonna:

```
insert into address(city_name)
values('City of South San Francisco'::char(20));
```

In questo caso, i primi 20 caratteri della stringa (`City of South San Fr`) verrebbero caricati nella colonna.

Tipi datetime

Argomenti

- Storage e intervalli

I tipi di dati datetime comprendono DATE, TIME, TIMETZ, TIMESTAMP e TIMESTAMPTZ.

Storage e intervalli

Nome	Storage	Intervallo	Risoluzione
DATE	4 byte	Da 4.713 BC a 294.276 AD	1 giorno
TIME	8 byte	Da 00:00:00 a 24:00:00	1 microsecondo
TIMETZ	8 byte	Da 00:00:00+1459 a 00:00:00+1459	1 microsecondo
TIMESTAMP	8 byte	Da 4.713 BC a 294.276 AD	1 microsecondo
TIMESTAMPTZ	8 byte	Da 4.713 BC a 294.276 AD	1 microsecondo

DATE

Utilizzare il tipo di dati DATE per memorizzare semplici date di calendario senza timestamp.

TIME

TIME è un alias di TIME WITHOUT TIME ZONE.

Utilizzare il tipo di dati TIME per memorizzare l'ora del giorno.

Le colonne TIME memorizzano valori con un massimo di 6 cifre di precisione per frazioni di secondo.

Per impostazione predefinita, i valori TIME sono in formato UTC sia nelle tabelle dell'utente sia nelle tabelle di sistema di Amazon Redshift.

TIMETZ

TIMETZ è un alias di TIME WITH TIME ZONE.

Utilizzare il tipo di dati TIMETZ per memorizzare l'ora del giorno con un fuso orario.

Le colonne TIMETZ memorizzano valori con un massimo di 6 cifre di precisione per frazioni di secondo.

Per impostazione predefinita, i valori TIMETZ sono UTC sia nelle tabelle dell'utente sia nelle tabelle di sistema Amazon Redshift.

TIMESTAMP

TIMESTAMP è un alias di TIMESTAMP WITHOUT TIME ZONE.

Utilizzare il tipo di dati TIMESTAMP per memorizzare valori timestamp completi che comprendono la data e l'ora del giorno.

Le colonne TIMESTAMP memorizzano valori fino a un massimo di 6 cifre di precisione per frazioni di secondo.

Se si inserisce una data in una colonna TIMESTAMP o una data con un valore timestamp parziale, il valore viene implicitamente convertito in un valore timestamp completo. Questo valore timestamp completo ha valori predefiniti (00) per le ore, i minuti e i secondi mancanti. I valori di fuso orario nelle stringhe input vengono ignorati.

Per impostazione predefinita, i valori TIMESTAMP sono UTC sia nelle tabelle dell'utente sia nelle tabelle di sistema Amazon Redshift.

TIMESTAMPTZ

TIMESTAMPTZ è un alias di TIMESTAMP WITH TIME ZONE.

Utilizzare il tipo di dati TIMESTAMPTZ per immettere valori timestamp completi che comprendono la data, l'ora del giorno e il fuso orario. Quando un valore di input include un fuso orario, Amazon Redshift usa il fuso orario per convertire il valore in formato UTC e memorizza il valore UTC.

Per visualizzare un elenco dei nomi di fuso orario supportati, utilizzare il comando seguente.

```
select pg_timezone_names();
```

Per visualizzare un elenco delle abbreviazioni di fuso orario supportate, utilizzare il comando seguente.

```
select pg_timezone_abbrevs();
```

È possibile trovare informazioni attuali sui fusi orari anche nel database dei fusi orari IANA.

La tabella seguente fornisce esempi di formati di fusi orari.

Formato	Esempio
gg mmm hh:mi:ss aaaa tz	17 Dic 07:37:16 1997 PST
mm/dd/yyyy hh:mi:ss.ss tz	12/17/1997 07:37:16.00 PST
mm/dd/yyyy hh:mi:ss.ss tz	12/17/1997 07:37:16.00 US/Pacific
yyyy-mm-dd hh:mi:ss+/-tz	1997-12-17 07:37:16-08
dd.mm.yyyy hh:mi:ss tz	17.12.1997 07:37:16.00 PST

Le colonne TIMESTAMPTZ memorizzano valori fino a un massimo di 6 cifre di precisione per frazioni di secondo.

Se si inserisce una data in una colonna TIMESTAMPTZ o una data con un valore timestamp parziale, il valore viene implicitamente convertito in un valore timestamp completo. Questo valore timestamp completo ha valori predefiniti (00) per le ore, i minuti e i secondi mancanti.

I valori TIMESTAMPTZ sono in formato UTC nelle tabelle utente.

Esempi con tipi datetime

Di seguito sono riportati degli esempi per lavorare con tipi datetime supportati da Amazon Redshift.

Esempi di data

Nei seguenti esempi vengono inserite date con formati diversi e viene visualizzato il risultato.

```
create table datetable (start_date date, end_date date);
```

```
insert into datetable values ('2008-06-01','2008-12-31');

insert into datetable values ('Jun 1,2008','20081231');
```

```
select * from datetable order by 1;

start_date |  end_date
-----------------------
2008-06-01 | 2008-12-31
2008-06-01 | 2008-12-31
```

Se si inserisce un valore timestamp in una colonna DATE, la parte dell'ora viene ignorata e viene caricata solo la data.

Esempi di orari

Negli esempi seguenti vengono inseriti valori TIME e TIMETZ con formati diversi e viene visualizzato il risultato.

```
create table timetable (start_time time, end_time timetz);
```

```
insert into timetable values ('19:11:19','20:41:19 UTC');
insert into timetable values ('191119', '204119 UTC');
```

```
select * from timetable order by 1;
start_time |  end_time
-----------------------
 19:11:19  | 20:41:19+00
 19:11:19  | 20:41:19+00
```

Esempi timestamp

Se inserisci una data in una colonna TIMESTAMP o TIMESTAMPTZ, l'ora diventa mezzanotte per impostazione predefinita. Ad esempio, se inserisci il valore letterale `20081231`, il valore memorizzato è `2008-12-31 00:00:00`.

Per modificare il fuso orario per la sessione corrente, usa il comando SET per impostare il parametro di configurazione timezone.

L'esempio seguente inserisce i timestamp con formati diversi e visualizza la tabella risultante.

```
create table tstamp(timeofday timestamp, timeofdaytz timestamptz);

insert into tstamp values('Jun 1,2008  09:59:59', 'Jun 1,2008 09:59:59 EST' );
insert into tstamp values('Dec 31,2008 18:20','Dec 31,2008 18:20');
insert into tstamp values('Jun 1,2008  09:59:59 EST', 'Jun 1,2008 09:59:59');

SELECT * FROM tstamp;

+---------------------+-----------------------+
|      timeofday      |      timeofdaytz      |
+---------------------+-----------------------+
| 2008-06-01 09:59:59 | 2008-06-01 14:59:59+00 |
| 2008-12-31 18:20:00 | 2008-12-31 18:20:00+00 |
| 2008-06-01 09:59:59 | 2008-06-01 09:59:59+00 |
+---------------------+-----------------------+
```

Valori letterali di data, ora e timestamp

Individuare le regole per lavorare con valori letterali data, ora e timestamp supportati da Amazon Redshift.

Date:

Le seguenti date di input sono tutti esempi validi di valori di data letterali per il tipo di dati DATE che puoi caricare nelle tabelle Amazon Redshift. Si assume che la modalità `MDY DateStyle` di default sia in vigore. Questa modalità indica che il valore del mese precede il valore del giorno in stringhe come `1999-01-08` e `01/02/00`.

> **Note**
>
> È necessario che un valore letterale data o timestamp quando viene caricato in una tabella sia racchiuso tra virgolette.

Data di input	Data completa
8 gennaio 1999	8 gennaio 1999
1999-01-08	8 gennaio 1999

Data di input	Data completa
1/8/1999	8 gennaio 1999
01/02/00	2 gennaio 2000
2000-Jan-31	31 gennaio 2000
Jan-31-2000	31 gennaio 2000
31-Jan-2000	31 gennaio 2000
20080215	15 febbraio 2008
080215	15 febbraio 2008
2008.366	31 dicembre 2008 (è necessario che la parte a 3 cifre della data sia compresa tra 001 e 366)

Volte

I seguenti orari di input sono tutti esempi validi di valori temporali letterali per i tipi di dati TIME e TIMETZ che puoi caricare nelle tabelle Amazon Redshift.

Input di orari	Descrizione (della parte dell'ora)
04:05:06.789	4:05 AM e 6.789 secondi
04:05:06	4:05 AM e 6 secondi
04:05	4:05 AM preciso
040506	4:05 AM e 6 secondi
04:05 AM	4:05 AM preciso; AM è facoltativo
04:05 PM	4:05 PM precise; è necessario che il valore dell'ora sia < 12.
16:05	4:05 PM preciso

Timestamp

I seguenti timestamp di input sono tutti esempi validi di valori temporali letterali per i tipi di dati TIMESTAMP e TIMESTAMPTZ che puoi caricare nelle tabelle Amazon Redshift. È possibile combinare tutti i valori letterali di data validi con i seguenti valori letterali di ora.

Timestamp di input (data e ora concatenate)	Descrizione (della parte dell'ora)
20080215 04:05:06.789	4:05 AM e 6.789 secondi
20080215 04:05:06	4:05 AM e 6 secondi
20080215 04:05	4:05 AM preciso
20080215 040506	4:05 AM e 6 secondi
20080215 04:05 AM	4:05 AM preciso; AM è facoltativo
20080215 04:05 PM	4:05 PM precise; è necessario che il valore dell'ora sia minore di 12.
20080215 16:05	4:05 PM preciso
20080215	Mezzanotte (per impostazione predefinita)

Valori datetime speciali

È possibile usare i valori speciali seguenti come valori letterali datetime e come argomenti per le funzioni della data. Richiedono virgolette singole e vengono convertiti in valori timestamp regolari durante l'elaborazione delle query.

Valore speciale	Descrizione
`now`	Valuta all'ora di inizio della transazione attuale e restituisce un timestamp con precisione di microsecondi.
`today`	Valuta alla data appropriata e restituisce un timestamp con più zeri al posto dell'ora.

Valore speciale	Descrizione
`tomorrow`	Valuta alla data appropriata e restituisce un timestamp con più zeri al posto dell'ora.
`yesterday`	Valuta alla data appropriata e restituisce un timestamp con più zeri al posto dell'ora.

I seguenti esempi mostrano come `now` e `today` lavorano insieme alla funzione DATEADD.

```
select dateadd(day,1,'today');

date_add
---------------------
2009-11-17 00:00:00
(1 row)

select dateadd(day,1,'now');

date_add
----------------------------
2009-11-17 10:45:32.021394
(1 row)
```

Tipi di dati e valori letterali a intervalli

Puoi utilizzare un tipo di dati a intervalli per memorizzare le durate di tempo in unità come,,,, `seconds` `minutes`, `hours` e. `days` `months` `years` I tipi di dati e i valori letterali degli intervalli possono essere utilizzati nei calcoli data/ora, ad esempio aggiungendo intervalli a date e timestamp, sommando intervalli e sottraendo un intervallo da una data o un timestamp. I valori letterali degli intervalli possono essere utilizzati come valori di input per intervallare le colonne dei tipi di dati in una tabella.

Sintassi del tipo di dati a intervalli

Per specificare un tipo di dati a intervalli per memorizzare una durata di tempo in anni e mesi:

```
INTERVAL year_to_month_qualifier
```

Per specificare un tipo di dati a intervalli per memorizzare una durata in giorni, ore, minuti e secondi:

```
INTERVAL day_to_second_qualifier [ (fractional_precision) ]
```

Sintassi dell'intervallo letterale

Per specificare un intervallo letterale per definire una durata di tempo in anni e mesi:

```
INTERVAL quoted-string year_to_month_qualifier
```

Per specificare un intervallo letterale per definire una durata in giorni, ore, minuti e secondi:

```
INTERVAL quoted-string day_to_second_qualifier [ (fractional_precision) ]
```

Argomenti

stringa tra virgolette

Specifica un valore numerico positivo o negativo che specifica una quantità e l'unità data/ora come stringa di input. Se la stringa tra virgolette contiene solo un numero, Amazon Redshift determina le unità da year_to_month_qualifier o day_to_second_qualifier. Ad `'23' MONTH` `1 year 11 months` `-2 days 0 hours 0 minutes 0.0 seconds` esempio `'-2' DAY` `'1-2' MONTH` `1 year 2 months`, `'13 day 1 hour 1 minute 1.123 seconds' SECOND` `13 days 1 hour 1 minute 1.123 seconds` rappresenta, rappresenta e rappresenta. Per ulteriori informazioni sui formati di output di un intervallo, vedereStili di intervallo.

qualificazione anno_per_mese

Speciifica l'intervallo dell'intervallo. Se usi un qualificatore e crei un intervallo con unità di tempo più piccole del qualificatore, Amazon Redshift tronca e scarta le parti più piccole dell'intervallo. I valori validi per year_to_month_qualifier sono:

- `YEAR`
- `MONTH`
- `YEAR TO MONTH`

day_to_second_qualifier

Speciifica l'intervallo dell'intervallo. Se usi un qualificatore e crei un intervallo con unità di tempo più piccole del qualificatore, Amazon Redshift tronca e scarta le parti più piccole dell'intervallo. I valori validi per day_to_second_qualifier sono:

- `DAY`

- HOUR
- MINUTE
- SECOND
- DAY TO HOUR
- DAY TO MINUTE
- DAY TO SECOND
- HOUR TO MINUTE
- HOUR TO SECOND
- MINUTE TO SECOND

L'output del valore letterale INTERVAL viene troncato al componente INTERVAL più piccolo specificato. Ad esempio, quando si utilizza un qualificatore MINUTE, Amazon Redshift elimina le unità di tempo inferiori a MINUTE.

```
select INTERVAL '1 day 1 hour 1 minute 1.123 seconds' MINUTE
```

Il valore risultante viene troncato a. `'1 day 01:01:00'`

precisione_frazionale

Parametro opzionale che specifica il numero di cifre frazionarie consentite nell'intervallo. L'argomento fractional_precision deve essere specificato solo se l'intervallo contiene SECOND. Ad esempio, `SECOND(3)` crea un intervallo che consente solo tre cifre frazionarie, ad esempio 1,234 secondi. Il numero massimo di cifre frazionarie è sei.

La configurazione della sessione `interval_forbid_composite_literals` determina se viene restituito un errore quando viene specificato un intervallo con le parti DA ANNO A MESE e DA GIORNO A SECONDO. Per ulteriori informazioni, consulta interval_forbid_composite_literals.

Aritmetica degli intervalli

È possibile utilizzare valori di intervallo con altri valori di data e ora per eseguire operazioni aritmetiche. La tabella seguente descrive le operazioni disponibili e il tipo di dati risultante da ciascuna operazione. Ad esempio, quando si aggiunge un `interval` a, `date` il risultato è un `date` se si tratta di un intervallo DA ANNO A MESE e un timestamp se si tratta di un intervallo DA UN GIORNO A UN SECONDO.

		Data	Timestamp	Interval	Numerico
Interval (Intervallo)	-	N/D	N/D	Interval	N/D
	+	Data	Data/ora	Interval	N/D
	*	N/D	N/D	N/D	Interval
	/	N/D	N/D	N/D	Interval
Data	-	Numerico	Interval	Data/ora	Data
	+	N/D	N/D	N/D	N/D
Time stamp	-	Interval	Interval	Timestamp	Timestamp
	+	N/D	N/D	N/D	N/D

Stili di intervallo

È possibile utilizzare il the section called “SET” comando SQL per modificare il formato di visualizzazione dell'output dei valori degli intervalli. Quando utilizzi il tipo di dati dell'intervallo in SQL, trasmettilo in testo per visualizzare lo stile di intervallo previsto, ad esempio. `YEAR TO MONTH::text` I valori disponibili per IMPOSTARE il `IntervalStyle` valore sono:

- `postgres`— segue lo stile PostgreSQL. Questa è l'impostazione predefinita.
- `postgres_verbose`— segue lo stile verboso di PostgreSQL.
- `sql_standard`— segue lo stile letterale a intervalli standard SQL.

Il comando seguente imposta lo stile dell'intervallo su. `sql_standard`

```
SET IntervalStyle to 'sql_standard';
```

formato di output postgres

Di seguito è riportato il formato di output per lo stile degli `postgres` intervalli. Ogni valore numerico può essere negativo.

```
'<numeric> <unit> [, <numeric> <unit> ...]'
```

```
select INTERVAL '1-2' YEAR TO MONTH::text

varchar
---------------
1 year 2 mons
```

```
select INTERVAL '1 2:3:4.5678' DAY TO SECOND::text

varchar
------------------
1 day 02:03:04.5678
```

formato di output postgres_verbose

La sintassi di postgres_verbose è simile a postgres, ma gli output di postgres_verbose contengono anche l'unità di tempo.

```
'[@] <numeric> <unit> [, <numeric> <unit> ...] [direction]'
```

```
select INTERVAL '1-2' YEAR TO MONTH::text

varchar
-----------------
@ 1 year 2 mons
```

```
select INTERVAL '1 2:3:4.5678' DAY TO SECOND::text

varchar
---------------------------
@ 1 day 2 hours 3 mins 4.56 secs
```

formato di output sql_standard

I valori dell'intervallo da anno a mese sono formattati come segue. Se si specifica un segno negativo prima dell'intervallo, si indica che l'intervallo è un valore negativo e si applica all'intero intervallo.

```
'[-]yy-mm'
```

I valori dell'intervallo da giorno a secondo sono formattati come segue.

```
'[-]dd hh:mm:ss.ffffff'
```

```
SELECT INTERVAL '1-2' YEAR TO MONTH::text

varchar
-------
1-2
```

```
select INTERVAL '1 2:3:4.5678' DAY TO SECOND::text

varchar
---------------
1 2:03:04.5678
```

Esempi di tipi di dati a intervalli

Gli esempi seguenti mostrano come utilizzare i tipi di dati INTERVAL con le tabelle.

```
create table sample_intervals (y2m interval month, h2m interval hour to minute);
insert into sample_intervals values (interval '20' month, interval '2 days
 1:1:1.123456' day to second);
select y2m::text, h2m::text from sample_intervals;


      y2m      |       h2m
---------------+-----------------
 1 year 8 mons | 2 days 01:01:00
```

```
update sample_intervals set y2m = interval '2' year where y2m = interval '1-8' year to
 month;
select * from sample_intervals;

   y2m   |       h2m
---------+-----------------
 2 years | 2 days 01:01:00
```

```
delete from sample_intervals where h2m = interval '2 1:1:0' day to second;
select * from sample_intervals;
```

```
 y2m | h2m
-----+-----
```

Esempi di valori letterali a intervalli

Gli esempi seguenti vengono eseguiti con lo stile dell'intervallo impostato su. `postgres`

L'esempio seguente mostra come creare un valore letterale INTERVAL di 1 anno.

```
select INTERVAL '1' YEAR

intervaly2m
---------------
1 years 0 mons
```

Se si specifica una stringa tra virgolette che supera il qualificatore, le unità di tempo rimanenti vengono troncate dall'intervallo. Nell'esempio seguente, un intervallo di 13 mesi diventa 1 anno e 1 mese, ma il restante 1 mese viene escluso a causa del qualificatore YEAR.

```
select INTERVAL '13 months' YEAR

intervaly2m
---------------
1 years 0 mons
```

Se si utilizza un qualificatore inferiore alla stringa di intervallo, vengono incluse le unità rimanenti.

```
select INTERVAL '13 months' MONTH

intervaly2m
---------------
1 years 1 mons
```

Se si specifica una precisione nell'intervallo, il numero di cifre frazionarie viene troncato alla precisione specificata.

```
select INTERVAL '1.234567' SECOND (3)

intervald2s
```

```
-------------------------------
0 days 0 hours 0 mins 1.235 secs
```

Se non specifichi una precisione, Amazon Redshift utilizza la precisione massima di 6.

```
select INTERVAL '1.23456789' SECOND

intervald2s
---------------------------------
0 days 0 hours 0 mins 1.234567 secs
```

L'esempio seguente mostra come creare un intervallo con intervalli.

```
select INTERVAL '2:2' MINUTE TO SECOND

intervald2s
-----------------------------
0 days 0 hours 2 mins 2.0 secs
```

I qualificatori determinano le unità che state specificando. Ad esempio, anche se l'esempio seguente utilizza la stessa stringa tra virgolette '2:2' dell'esempio precedente, Amazon Redshift riconosce che utilizza unità di tempo diverse a causa del qualificatore.

```
select INTERVAL '2:2' HOUR TO MINUTE

intervald2s
-----------------------------
0 days 2 hours 2 mins 0.0 secs
```

Sono supportate anche le abbreviazioni e i plurali di ciascuna unità. Ad esempio, `5s` `5 second`, e `5 seconds` sono intervalli equivalenti. Le unità supportate sono anni, mesi, ore, minuti e secondi.

```
select INTERVAL '5s' SECOND

intervald2s
-----------------------------
0 days 0 hours 0 mins 5.0 secs
```

```
select INTERVAL '5 HOURS' HOUR
```

```
intervald2s
----------------------------
0 days 5 hours 0 mins 0.0 secs
```

```
select INTERVAL '5 h' HOUR

intervald2s
----------------------------
0 days 5 hours 0 mins 0.0 secs
```

Esempi di valori letterali a intervalli senza sintassi dei qualificatori

> **Note**
>
> Negli esempi seguenti viene illustrato l'utilizzo di un valore letterale a intervalli senza un qualificatore or. `YEAR TO MONTH` `DAY TO SECOND` Per informazioni sull'utilizzo del valore letterale di intervallo consigliato con un qualificatore, vedere. Tipi di dati e valori letterali a intervalli

Usa un valore letterale di intervallo per identificare periodi di tempo specifici, come `12 hours` o `6 months`. È possibile usare questi valori letterali di intervallo in condizioni e calcoli che comprendono espressioni datetime.

Un valore letterale di intervallo viene espresso come combinazione della parola chiave INTERVAL con una quantità numerica e una parte di data supportata, ad esempio o. `INTERVAL '7 days'` `INTERVAL '59 minutes'` È possibile collegare diverse quantità e unità per formare un intervallo più preciso; ad esempio `INTERVAL '7 days, 3 hours, 59 minutes'`. Anche abbreviazioni e plurali di ciascuna unità sono supportati; ad esempio: `5 s`, `5 second` e `5 seconds` sono intervalli equivalenti.

Se non si specifica una parte data, il valore di intervallo rappresenterà i secondi. È possibile specificare il valore di quantità come una frazione (ad esempio: `0.5 days`).

Gli esempi seguenti mostrano una serie di calcoli con valori di intervallo diversi.

Quanto segue aggiunge 1 secondo alla data specificata.

```
select caldate + interval '1 second' as dateplus from date
```

```
where caldate='12-31-2008';
dateplus
---------------------
2008-12-31 00:00:01
(1 row)
```

Quanto segue aggiunge 1 minuto alla data specificata.

```
select caldate + interval '1 minute' as dateplus from date
where caldate='12-31-2008';
dateplus
---------------------
2008-12-31 00:01:00
(1 row)
```

Vengono aggiunte 3 ore e 35 minuti alla data specificata.

```
select caldate + interval '3 hours, 35 minutes' as dateplus from date
where caldate='12-31-2008';
dateplus
---------------------
2008-12-31 03:35:00
(1 row)
```

Quanto segue aggiunge 52 settimane alla data specificata.

```
select caldate + interval '52 weeks' as dateplus from date
where caldate='12-31-2008';
dateplus
---------------------
2009-12-30 00:00:00
(1 row)
```

Vengono aggiunti 1 settimana, 1 ora, 1 minuto e 1 secondo alla data specificata.

```
select caldate + interval '1w, 1h, 1m, 1s' as dateplus from date
where caldate='12-31-2008';
dateplus
---------------------
2009-01-07 01:01:01
(1 row)
```

Di seguito vengono aggiunte 12 ore (mezza giornata) alla data specificata.

```
select caldate + interval '0.5 days' as dateplus from date
where caldate='12-31-2008';
dateplus
---------------------
2008-12-31 12:00:00
(1 row)
```

Vengono sottratti 4 mesi dal 15 febbraio 2023 e il risultato è il 15 ottobre 2022.

```
select date '2023-02-15' - interval '4 months';

?column?
---------------------
2022-10-15 00:00:00
```

Vengono sottratti 4 mesi dal 31 marzo 2023 e il risultato è il 30 novembre 2022. Il calcolo considera il numero di giorni in un mese.

```
select date '2023-03-31' - interval '4 months';

?column?
---------------------
2022-11-30 00:00:00
```

Tipo booleano

Usa valori di dati BOOLEAN per memorizzare valori true e false in una colonna a byte singolo. La tabella seguente descrive i tre possibili stati per un valore booleano e i valori letterali che risultano in quello stato. Indipendentemente dalla stringa di input, una colonna booleana memorizza e restituisce "t" per true e "f" per false.

Stato	Valori letterali validi	Storage
True	`TRUE 't' 'true' 'y' 'yes' '1'`	1 byte

Stato	Valori letterali validi	Storage
False	`FALSE 'f' 'false' 'n' 'no' '0'`	1 byte
Sconosciuto	`NULL`	1 byte

È possibile utilizzare un confronto IS per controllare il valore Boolean solo come predicato nella clausola WHERE. Non è possibile utilizzare un confronto IS con un valore Boolean nell'elenco SELECT.

Esempi

È possibile usare una colonna BOOLEAN per memorizzare uno stato "Attivo/Inattivo" per ciascun cliente in una tabella CUSTOMER.

```
create table customer(
custid int,
active_flag boolean default true);
```

```
insert into customer values(100, default);
```

```
select * from customer;
custid | active_flag
-------+-------------
   100 | t
```

Se non viene specificato alcun valore predefinito (`true` o `false`) nell'istruzione CREATE TABLE, inserire un valore predefinito significa inserire un valore null.

In questo esempio, la query seleziona dalla tabella USERS utenti a cui piacciono gli sport ma non il teatro:

```
select firstname, lastname, likesports, liketheatre
from users
where likesports is true and liketheatre is false
```

```
order by userid limit 10;

firstname |  lastname  | likesports | liketheatre
----------+------------+------------+-------------
Lars      | Ratliff    | t          | f
Mufutau   | Watkins    | t          | f
Scarlett  | Mayer      | t          | f
Shafira   | Glenn      | t          | f
Winifred  | Cherry     | t          | f
Chase     | Lamb       | t          | f
Liberty   | Ellison    | t          | f
Aladdin   | Haney      | t          | f
Tashya    | Michael    | t          | f
Lucian    | Montgomery | t          | f
(10 rows)
```

Il seguente esempio seleziona dalla tabella USERS gli utenti per i quali non si sa se gradiscono la musica rock.

```
select firstname, lastname, likerock
from users
where likerock is unknown
order by userid limit 10;

firstname | lastname | likerock
----------+----------+----------
Rafael    | Taylor   |
Vladimir  | Humphrey |
Barry     | Roy      |
Tamekah   | Juarez   |
Mufutau   | Watkins  |
Naida     | Calderon |
Anika     | Huff     |
Bruce     | Beck     |
Mallory   | Farrell  |
Scarlett  | Mayer    |
(10 rows)
```

L'esempio seguente restituisce un errore perché utilizza un confronto IS nell'elenco SELECT.

```
select firstname, lastname, likerock is true as "check"
from users
order by userid limit 10;
```

```
[Amazon](500310) Invalid operation: Not implemented
```

L'esempio seguente ha esito positivo perché utilizza un confronto uguale (=) nell'elenco SELECT invece del confronto IS.

```
select firstname, lastname, likerock = true as "check"
from users
order by userid limit 10;

firstname | lastname  | check
----------+-----------+------
Rafael    | Taylor    |
Vladimir  | Humphrey  |
Lars      | Ratliff   | true
Barry     | Roy       |
Reagan    | Hodge     | true
Victor    | Hernandez | true
Tamekah   | Juarez    |
Colton    | Roy       | false
Mufutau   | Watkins   |
Naida     | Calderon  |
```

Tipo HLLSKETCH

Utilizzate il tipo di dati HLLSKETCH per gli schizzi. HyperLogLog Amazon Redshift supporta rappresentazioni di HyperLogLog schizzi sparse o dense. Gli schizzi iniziano come sparsi e passano a densi quando il formato denso è più efficiente per ridurre al minimo l'ingombro di memoria utilizzato.

Amazon Redshift esegue automaticamente la transizione di uno HyperLogLog schizzo sparso durante l'importazione, l'esportazione o la stampa di schizzi nel seguente formato JSON.

```
{"logm":15,"sparse":{"indices":[4878,9559,14523],"values":[1,2,1]}}
```

Amazon Redshift utilizza una rappresentazione di stringa in formato Base64 per rappresentare uno sketch denso. HyperLogLog

Amazon Redshift utilizza la seguente rappresentazione di stringa in formato Base64 per rappresentare uno sketch denso. HyperLogLog

```
"ABAABA..."
```

La dimensione massima di un oggetto HLLSKETCH è di 24.580 byte se utilizzato nella compressione raw.

Tipo SUPER

Utilizzare il tipo di dati SUPER per memorizzare documenti o dati semistrutturati come valori.

I dati semistrutturati non sono conformi alla struttura rigida e tabulare del modello di dati relazionali utilizzato nei database SQL. Contiene tag che fanno riferimento a entità distinte all'interno dei dati. Possono contenere valori complessi quali array, strutture nidificate e altre strutture complesse associate ai formati di serializzazione, ad esempio JSON. Il tipo di dati SUPER è un insieme di valori di struttura e array senza schema che comprendono tutti gli altri tipi scalari di Amazon Redshift.

Il tipo di dati SUPER supporta fino a 16 MB di dati per un singolo oggetto SUPER. Per ulteriori informazioni sul tipo di dati SUPER, con esempi di implementazione in una tabella, consulta Importazione e query di dati semistrutturati in Amazon Redshift.

Gli oggetti SUPER di dimensioni superiori a 1 MB possono essere importati solo dai seguenti formati di file:

- Parquet
- JSON
- TEXT
- CSV

Il tipo di dati SUPER presenta le seguenti proprietà:

- Un valore scalare Amazon Redshift:
  - Un valore null
  - Un valore booleano
  - Un numero, ad esempio smallint, integer, bigint, decimale o virgola mobile (ad esempio float4 o float8)
  - Un valore di stringa, ad esempio varchar o char
- Un valore complesso:
  - Un array di valori, inclusi scalari o complessi
  - Una struttura, nota anche come tupla o oggetto, ovvero una mappa di nomi e valori degli attributi (scalari o complessi)

Uno qualsiasi dei due tipi di valori complessi contiene i propri scalari o valori complessi senza avere alcuna limitazione sulla regolarità.

Il tipo di dati SUPER supporta la persistenza di dati semistrutturati in un formato senza schema. Anche se il modello di dati gerarchico può cambiare, le vecchie versioni dei dati possono coesistere nella stessa colonna SUPER.

Amazon Redshift utilizza PartiQL per abilitare la navigazione in array e strutture. Amazon Redshift utilizza la sintassi PartiQL anche per eseguire l'iterazione su array SUPER. Per ulteriori informazioni, consulta Navigazione e Annullamento di query.

Amazon Redshift utilizza la digitazione dinamica per elaborare dati SUPER senza schema e non dichiara i tipi di dati prima di utilizzarli nella query. Per ulteriori informazioni, consulta Digitazione dinamica.

È possibile applicare le politiche di mascheramento dei dati dinamici ai valori `scalar` sui percorsi delle colonne di tipo SUPER. Per ulteriori informazioni sul mascheramento dei dati dinamici, consulta Mascheramento dinamico dei dati. Per informazioni su come usare il mascheramento dei dati dinamici con il tipo di dati SUPER, consulta Utilizzo del mascheramento dei dati dinamici con percorsi di tipo di dati SUPER.

Tipo VARBYTE

Usa una colonna VARBYTE, VARBINARY o BINARY VARYING per memorizzare il valore binario di lunghezza variabile con un limite fisso.

```
varbyte [ (n) ]
```

Il numero massimo di byte (n) può variare da 1 a 16.777.216. Il valore di default è 64.000.

Alcuni esempi in cui è possibile utilizzare un tipo di dati VARBYTE sono i seguenti:

- Unire le tabelle sulle colonne VARBYTE.
- Creazione di viste materializzate che contengono colonne VARBYTE. È supportato l'aggiornamento incrementale delle viste materializzate che contengono colonne VARBYTE. Tuttavia, le funzioni aggregate diverse da COUNT, MIN e MAX e GROUP BY sulle colonne VARBYTE non supportano l'aggiornamento incrementale.

Per garantire che tutti i byte siano caratteri stampabili, Amazon Redshift utilizza il formato esadecimale per stampare i valori VARBYTE. Ad esempio, il seguente SQL converte la stringa

esadecimale 6162 in un valore binario. Anche se il valore restituito è un valore binario, i risultati vengono stampati come esadecimale 6162.

```
select from_hex('6162');

 from_hex
----------
 6162
```

Amazon Redshift supporta il casting tra VARBYTE e i seguenti tipi di dati:

- CHAR
- VARCHAR
- SMALLINT
- INTEGER
- BIGINT

Quando si esegue il casting con CHAR e VARCHAR viene utilizzato il formato UTF-8. Per ulteriori informazioni sul formato UTF-8, consulta TO_VARBYTE. Quando si esegue il casting da SMALLINT, INTEGER e BIGINT, viene mantenuto il numero di byte del tipo di dati originale. Cioè due byte per SMALLINT, quattro byte per INTEGER e otto byte per BIGINT.

La seguente istruzione SQL lancia una stringa VARCHAR in un VARBYTE. Anche se il valore restituito è un valore binario, i risultati vengono stampati come esadecimale 616263.

```
select 'abc'::varbyte;

 varbyte
---------
 616263
```

L'istruzione SQL seguente lancia un valore CHAR in una colonna in un VARBYTE. In questo esempio viene creata una tabella con una colonna CHAR (10) (c), inserisce valori di carattere più brevi della lunghezza di 10. Il cast risultante associa il risultato con caratteri spaziali (hex'20') alla dimensione della colonna definita. Anche se il valore restituito è un valore binario, i risultati vengono stampati come esadecimale.

```
create table t (c char(10));
```

```
insert into t values ('aa'), ('abc');
select c::varbyte from t;
          c
----------------------
 6161202020202020202020
 6162632020202020202020
```

La seguente istruzione SQL lancia una stringa SMALLINT in un VARBYTE. Anche se il valore restituito è un valore binario, i risultati vengono stampati come esadecimale `0005`, ovvero due byte o quattro caratteri esadecimali.

```
select 5::smallint::varbyte;

 varbyte
---------
 0005
```

La seguente istruzione SQL lancia un INTEGER in un VARBYTE. Anche se il valore restituito è un valore binario, i risultati vengono stampati come esadecimale `00000005`, ovvero quattro byte o otto caratteri esadecimali.

```
select 5::int::varbyte;

 varbyte
----------
 00000005
```

La seguente istruzione SQL lancia un BIGINT in un VARBYTE. Anche se il valore restituito è un valore binario, i risultati vengono stampati come esadecimale `0000000000000005`, ovvero otto byte o 16 caratteri esadecimali.

```
select 5::bigint::varbyte;

     varbyte
------------------
 0000000000000005
```

Le funzionalità di Amazon Redshift che supportano il tipo di dati VARBYTE includono:

- Operatori VARBYTE

- CONCAT
- LEN
- Funzione LENGTH
- OCTET_LENGTH
- Funzione SUBSTRING
- FROM_HEX
- TO_HEX
- FROM_VARBYTE
- TO_VARBYTE
- GETBIT
- Caricamento di una colonna definita come tipo dati VARBYTE
- Scaricare una colonna definita come tipo dati VARBYTE

Limitazioni nell'utilizzo del tipo di dati VARBYTE con Amazon Redshift

Di seguito sono elencale le limitazioni nell'utilizzo del tipo di dati VARBYTE con Amazon Redshift:

- Amazon Redshift Spectrum supporta il tipo di dati VARBYTE solo per i file Parquet e ORC.
- L'editor di query Amazon Redshift e l'editor di query Amazon Redshift v2 non supportano ancora completamente il tipo di dati VARBYTE. Pertanto, utilizzare un client SQL diverso quando si lavora con espressioni VARBYTE.

  Come soluzione alternativa per utilizzare l'editor di query, se la lunghezza dei dati è inferiore a 64 KB e il contenuto è valido UTF-8, è possibile eseguire il cast dei valori VARBYTE su VARCHAR, ad esempio:

  ```
  select to_varbyte('6162', 'hex')::varchar;
  ```
- Non è possibile utilizzare i tipi di dati VARBYTE con le funzioni definite dall'utente (UDF) Python o Lambda.
- Non è possibile creare una colonna HLLSKETCH da una colonna VARBYTE o utilizzare APPROSSIMATE COUNT DISTINCT su una colonna VARBYTE.
- I valori VARBYTE superiori a 1 MB possono essere importati solo dai seguenti formati di file:
  - Parquet
  - Testo

- Valori separati da virgole (CSV)

Conversione e compatibilità dei tipi

Di seguito è riportata una discussione su come le regole di conversione dei tipi e la compatibilità dei tipi di dati funzionano in Amazon Redshift.

Compatibilità

La corrispondenza dei tipi di dati e la corrispondenza di valori letterali e costanti con tipi di dati avviene durante diverse operazioni di database, comprese le seguenti:

- Operazioni DML (Data Manipulation Language) sulle tabelle
- Query UNION, INTERSECT ed EXCEPT
- Espressioni CASE
- Valutazione di predicati, come LIKE e IN
- Valutazione di funzioni SQL che effettuano confronti o estrazioni di dati
- Confronti con operatori matematici

I risultati di queste operazioni dipendono dalle regole di conversione dei tipi e dalla compatibilità dei tipi di dati. La compatibilità implica che non è one-to-one sempre richiesta la corrispondenza di un determinato valore e di un determinato tipo di dati. Dato che alcuni tipi di dati sono compatibili, una conversione implicita o coercizione può essere effettuata (per maggiori informazioni, consultare Tipi di conversione implicita). Quando i tipi di dati non sono compatibili, a volte è possibile convertire un valore da un tipo di dati a un altro usando una funzione di conversione esplicita.

Regole generali di conversione e compatibilità

Osserva le seguenti regole di conversione e compatibilità:

- In generale, i tipi di dati che rientrano nella stessa categoria (come diversi tipi di dati numerici) sono compatibili ed è possibile convertirli in modo implicito.

  Ad esempio, con la conversione implicita è possibile inserire un valore decimale in una colonna intera. Il decimale viene arrotondato per produrre un numero intero. Altrimenti, è possibile estrarre un valore numerico, come `2008`, da una data e inserirlo nella colonna intera.
- I tipi di dati numerici applicano le condizioni di overflow che si verificano quando si tenta di inserire valori. out-of-range Ad esempio, un valore decimale con una precisione di 5 non rientra in una

colonna decimale che è stata definita con una precisione di 4. Un intero o la parte intera di un decimale non viene mai troncata; tuttavia, è possibile arrotondare la parte frazionaria di un decimale per difetto o per eccesso, come appropriato. Tuttavia, i risultati di espliciti cast di valori selezionati dalle tabelle non sono arrotondati.

- Diversi tipi di stringhe di caratteri sono compatibili; le stringhe colonna VARCHAR contenenti dati a byte singolo e le stringhe colonna CHAR sono confrontabili e convertibili in modo implicito. Le stringhe VARCHAR che contengono dati multibyte non sono confrontabili. Inoltre, è possibile convertire una stringa di caratteri in un valore numerico, timestamp o data se la stringa è un valore letterale appropriato; eventuali spazi iniziali o finali vengono ignorati. Per contro, è possibile convertire un valore numero, timestamp o data in una stringa di caratteri a lunghezza variabile o fissa.

  > **Note**
  >
  > È necessario che una stringa di caratteri per la quale si desidera eseguire il cast a un tipo numerico contenga una rappresentazione in caratteri di un numero. Ad esempio, è possibile eseguire il cast per le stringhe `'1.0'` o `'5.9'` a valori decimali, ma non è possibile eseguire il cast per la stringa `'ABC'` a nessun tipo numerico.

- Se si confrontano i valori DECIMAL con le stringhe di caratteri, Amazon Redshift tenta di convertire la stringa di caratteri in un valore DECIMAL. Quando si confrontano tutti gli altri valori numerici con stringhe di caratteri, i valori numerici vengono convertiti in stringhe di caratteri. Per applicare la conversione opposta (ad esempio, convertire le stringhe di caratteri in numeri interi o convertire i valori DECIMAL in stringhe di caratteri), usa una funzione esplicita, ad esempio, CAST.
- Per convertire valori DECIMAL o NUMERIC a 64 bit in una precisione più elevata, è necessario usare una funzione di conversione specifica come le funzioni CAST o CONVERT.
- Quando si convertono DATE o TIMESTAMP in TIMESTAMPTZ o si converte TIME in TIMESTAMPTZ, il fuso orario viene impostato sul fuso orario della sessione attuale. Il fuso orario della sessione è UTC per impostazione predefinita. Per ulteriori informazioni sulle impostazioni del fuso orario della sessione, consultare timezone.
- Allo stesso modo, TIMESTAMPTZ viene convertito in DATE, TIME o TIMESTAMP sulla base del fuso orario della sessione corrente. Il fuso orario della sessione è UTC per impostazione predefinita. Dopo la conversione, le informazioni sul fuso orario vengono rimosse.
- Le stringhe di caratteri che rappresentano un timestamp con fuso orario specificato vengono convertite in TIMESTAMPTZ usando il fuso orario della sessione corrente, che di default è UTC. Analogamente, le stringhe di caratteri che rappresentano un tempo con fuso orario specificato

vengono convertite in TIMETZ usando il fuso orario della sessione attuale, che per impostazione predefinita è UTC.

Tipi di conversione implicita

Ci sono due tipi di conversione implicita:

- Conversioni implicite negli incarichi, come i valori delle impostazioni nei comandi INSERT o UPDATE.
- Conversioni implicite nelle espressioni, come l'esecuzione di confronti nella clausola WHERE.

La tabella seguente elenca i tipi di dati che è possibile convertire in modo implicito negli incarichi o nelle espressioni. È anche possibile usare una funzione di conversione esplicita per eseguire queste conversioni.

Dal tipo	Al tipo
BIGINT (INT8)	BOOLEAN
	CHAR
	DECIMAL (NUMERIC)
	DOUBLE PRECISION (FLOAT8)
	INTEGER (INT, INT4)
	REAL (FLOAT4)
	SMALLINT (INT2)
	VARCHAR
CHAR	VARCHAR
DATE	CHAR
	VARCHAR
	TIMESTAMP

Dal tipo	Al tipo
	TIMESTAMPTZ
DECIMAL (NUMERIC)	BIGINT (INT8)
	CHAR
	DOUBLE PRECISION (FLOAT8)
	INTEGER (INT, INT4)
	REAL (FLOAT4)
	SMALLINT (INT2)
	VARCHAR
DOUBLE PRECISION (FLOAT8)	BIGINT (INT8)
	CHAR
	DECIMAL (NUMERIC)
	INTEGER (INT, INT4)
	REAL (FLOAT4)
	SMALLINT (INT2)
	VARCHAR
INTEGER (INT, INT4)	BIGINT (INT8)
	BOOLEAN
	CHAR
	DECIMAL (NUMERIC)
	DOUBLE PRECISION (FLOAT8)

Dal tipo	Al tipo
	REAL (FLOAT4)
	SMALLINT (INT2)
	VARCHAR
REAL (FLOAT4)	BIGINT (INT8)
	CHAR
	DECIMAL (NUMERIC)
	INTEGER (INT, INT4)
	SMALLINT (INT2)
	VARCHAR
SMALLINT (INT2)	BIGINT (INT8)
	BOOLEAN
	CHAR
	DECIMAL (NUMERIC)
	DOUBLE PRECISION (FLOAT8)
	INTEGER (INT, INT4)
	REAL (FLOAT4)
	VARCHAR
TIMESTAMP	CHAR
	DATE
	VARCHAR

Dal tipo	Al tipo
	TIMESTAMPTZ
	TIME
TIMESTAMPTZ	CHAR
	DATE
	VARCHAR
	TIMESTAMP
	TIMETZ
TIME	VARCHAR
	TIMETZ
	INTERVAL DAY TO SECOND
TIMETZ	VARCHAR
	TIME
GEOMETRY	GEOGRAPHY
GEOGRAPHY	GEOMETRY

Note

Le conversioni implicite tra TIMESTAMPTZ, TIMESTAMP, DATE, TIME, TIMETZ o stringhe di caratteri usano il fuso orario della sessione corrente. Per informazioni sulle impostazioni del fuso orario corrente, consultare timezone.

I tipi di dati GEOMETRY e GEOGRAPHY non possono essere convertiti implicitamente in nessun altro tipo di dati, eccetto l'un l'altro. Per ulteriori informazioni, consulta Funzione CAST.

Il tipo di dati VARBYTE non può essere convertito implicitamente in nessun altro tipo di dati. Per ulteriori informazioni, consulta Funzione CAST.

Utilizzo della digitazione dinamica per il tipo di dati SUPER

Amazon Redshift utilizza la digitazione dinamica per elaborare i dati SUPER senza schema senza la necessità di dichiarare i tipi di dati prima di utilizzarli nella query. La digitazione dinamica utilizza i risultati della navigazione nelle colonne di dati SUPER senza doverne eseguire esplicitamente il casting nei tipi Amazon Redshift. Per ulteriori informazioni su come usare la digitazione dinamica per il tipo di dati SUPER, consultare Digitazione dinamica.

È possibile eseguire il casting di valori SUPER da e verso altri tipi di dati con alcune eccezioni. Per ulteriori informazioni, consultare Limitazioni.

Sequenze di regole di confronto

Amazon Redshift non supporta sequenze di regole di confronto definite dall'utente o specifiche in termini di impostazioni locali. In generale, i risultati di qualsiasi predicato in qualsiasi contesto potrebbero essere influenzati dalla mancanza di regole specifiche in termini di impostazioni locali per l'ordinamento e il confronto di valori di dati. Ad esempio, le funzioni e le espressioni ORDER BY come MIN, MAX e RANK restituiscono risultati sulla base di un ordinamento UTF-8 binario dei dati che non prende in considerazione caratteri specifici in termini di impostazioni locali.

Espressioni

Argomenti

- Espressioni semplici
- Espressioni composte
- Elenco di espressioni
- Subquery scalari
- Espressioni di funzioni

Un'espressione è una combinazione di uno o più valori, operatori o funzioni che restituisce un valore. Il tipo di dati di un'espressione è solitamente quello dei suoi componenti.

Espressioni semplici

Un'espressione semplice è una delle seguenti:

- Una costante o un valore letterale
- Un nome o un riferimento di colonna
- Una funzione scalare
- Una funzione (set) di aggregazione
- Una funzione finestra
- Una subquery scalare

Esempi di espressioni semplici comprendono:

```
5+12
dateid
sales.qtysold * 100
sqrt (4)
max (qtysold)
(select max (qtysold) from sales)
```

Espressioni composte

Un'espressione composta è una serie di espressioni semplici unite da operatori aritmetici. È necessario che un'espressione semplice usata in un'espressione composta restituisca un valore numerico.

Sintassi

```
expression
operator
expression | (compound_expression)
```

Argomenti

espressione

Un'espressione semplice che restituisce un valore.

operatore

È possibile costruire un'espressione aritmetica composta usando gli operatori seguenti, in questo ordine di precedenza:

- ( ) : parentesi per controllare l'ordine di valutazione
- + , - : operatori/segni positivi e negativi
- ^ , |/ , ||/ : potenza, radice quadrata, radice cubica
- * , / , % : operatori di moltiplicazione, divisione e modulo
- @ : valore assoluto
- + , - : addizione e sottrazione
- & , |, #, ~, <<, >> : operatori bit per bit AND, OR, NOT, shift a sinistra, shift a destra
- ||: concatenazione

(compound_expression)

Le espressioni composte possono essere nidificate utilizzando le parentesi.

Esempi

Esempi di espressioni composte comprendono:

```
('SMITH' || 'JONES')
sum(x) / y
sqrt(256) * avg(column)
rank() over (order by qtysold) / 100
(select (pricepaid - commission) from sales where dateid = 1882) * (qtysold)
```

È anche possibile nidificare alcune funzioni in altre funzioni. Ad esempio, è possibile nidificare qualsiasi funzione scalare dentro un'altra funzione scalare. Nell'esempio seguente viene restituita la somma dei valori assoluti di un set di numeri:

```
sum(abs(qtysold))
```

Non è possibile usare le funzioni finestra come argomenti per le funzioni di aggregazione o altre funzioni finestra. L'espressione seguente restituirebbe un errore:

```
avg(rank() over (order by qtysold))
```

Le funzioni finestra possono avere una funzione di aggregazione nidificata. L'espressione seguente addiziona set di valori e successivamente li classifica:

```
rank() over (order by sum(qtysold))
```

Elenco di espressioni

Un elenco di espressioni è una combinazione di espressioni e può comparire nelle condizioni di confronto e appartenenza (clausole WHERE) e in clausole GROUP BY.

Sintassi

```
expression , expression , ... | (expression, expression, ...)
```

Argomenti

espressione

Un'espressione semplice che restituisce un valore. Un elenco di espressioni può contenere più espressioni separate da virgola o uno o più set di espressioni separate da virgola. Quando sono presenti più set di espressioni, è necessario che ciascun set contenga lo stesso numero di espressioni e sia separato da parentesi. È necessario che il numero di espressioni in ciascun set corrisponda al numero di espressioni prima dell'operatore nella condizione.

Esempi

Di seguito vengono riportati esempi di elenchi di espressioni in condizioni:

```
(1, 5, 10)
('THESE', 'ARE', 'STRINGS')
(('one', 'two', 'three'), ('blue', 'yellow', 'green'))
```

È necessario che il numero di espressioni in ciascun set corrisponda al numero nella prima parte dell'istruzione:

```
select * from venue
where (venuecity, venuestate) in (('Miami', 'FL'), ('Tampa', 'FL'))
order by venueid;
```

```
venueid |        venuename        | venuecity | venuestate | venueseats
---------+-------------------------+-----------+------------+------------
28 | American Airlines Arena | Miami     | FL         |          0
54 | St. Pete Times Forum    | Tampa     | FL         |          0
91 | Raymond James Stadium   | Tampa     | FL         |      65647
(3 rows)
```

Subquery scalari

Una subquery scalare è una query SELECT regolare tra parentesi che restituisce esattamente un valore: una riga con una colonna. La query viene eseguita e il valore restituito viene usato nella query outer. Se la subquery restituisce zero righe, il valore dell'espressione della subquery è null. Se restituisce più di una riga, Amazon Redshift restituisce un errore. La subquery può riferirsi a variabili della query padre, che agirà come costante durante qualsiasi invocazione della subquery.

È possibile usare le subquery scalari nella maggior parte delle istruzioni che chiamano un'espressione. Le subquery scalari non sono espressioni valide nei casi seguenti:

- Come valori predefiniti per espressioni
- Nelle clausole GROUP BY e HAVING

Esempio

La subquery seguente calcola il prezzo medio pagato per vendita nel corso dell'intero anno 2008, quindi la query outer usa quel valore nell'output per confrontarlo rispetto al prezzo medio per vendita per trimestre:

```
select qtr, avg(pricepaid) as avg_saleprice_per_qtr,
(select avg(pricepaid)
from sales join date on sales.dateid=date.dateid
where year = 2008) as avg_saleprice_yearly
from sales join date on sales.dateid=date.dateid
where year = 2008
group by qtr
order by qtr;
qtr  | avg_saleprice_per_qtr | avg_saleprice_yearly
-------+-----------------------+----------------------
1     |                647.64 |               642.28
2     |                646.86 |               642.28
3     |                636.79 |               642.28
4     |                638.26 |               642.28
```

```
(4 rows)
```

Espressioni di funzioni

Sintassi

Qualsiasi funzione integrata può essere usata come espressione. La sintassi per una chiamata di funzione è il nome di una funzione seguito dal suo elenco di argomenti tra parentesi.

```
function ( [expression [, expression...]] )
```

Argomenti

funzione

Qualsiasi funzione integrata. Per alcuni esempi di funzioni, consulta Informazioni di riferimento sulle funzioni SQL.

espressione

Qualsiasi espressione che corrisponda al numero di parametri e tipi di dati previsto dalla funzione.

Esempi

```
abs (variable)
select avg (qtysold + 3) from sales;
select dateadd (day,30,caldate) as plus30days from date;
```

Condizioni

Argomenti

- Sintassi
- Condizione di confronto
- Condizioni logiche
- Condizioni di corrispondenza di modelli
- Condizione di intervallo BETWEEN
- Condizione Null
- Condizione EXISTS

- Condizione IN

Una condizione è un'istruzione di una o più espressioni e operatori logici che restituisce un valore true, false o sconosciuto. A volte le condizioni vengono anche chiamate predicati.

> **Note**
>
> Tutte le corrispondenze tra modelli LIKE e i confronti di stringhe fanno distinzione tra maiuscole e minuscole. Ad esempio, "A" e "a" non corrispondono. Tuttavia, è possibile effettuare una corrispondenza tra modelli che non distingue tra maiuscole e minuscole usando il predicato ILIKE.

Sintassi

```
comparison_condition
| logical_condition
| range_condition
| pattern_matching_condition
| null_condition
| EXISTS_condition
| IN_condition
```

Condizione di confronto

Le condizioni di confronto esprimono le relazioni logiche tra due valori. Tutte le condizioni di confronto sono operatori binari con un tipo restituito booleano. Amazon Redshift supporta gli operatori di confronto descritti nella tabella seguente:

Operatore	Sintassi	Descrizione
<	a < b	Il valore a è inferiore al valore b.
>	a > b	Il valore a è superiore al valore b.
<=	a <= b	Il valore a è inferiore o uguale al valore b.
>=	a >= b	Il valore a è superiore o uguale al valore b.

Operatore	Sintassi	Descrizione
=	`a = b`	Il valore a è uguale al valore b.
<> o !=	`a <> b or a != b`	Il valore a non è uguale al valore b.
ANY | SOME	`a = ANY(subquery)`	Il valore a è uguale a qualsiasi valore restituito dalla subquery.
ALL	`a <> ALL or != ALL (subquery))`	Il valore a non è uguale a nessun valore restituito dalla subquery.
IS TRUE | FALSE | UNKNOWN	`a IS TRUE`	Il valore a è un valore booleano TRUE.

Note per l'utilizzo

= ANY | SOME

Le parole chiave ANY e SOME sono sinonimi della condizione IN e restituiscono true se il confronto ha esito positivo per almeno un valore restituito da una query secondaria che restituisce uno o più valori. Amazon Redshift supporta solo la condizione = (uguale) per ANY e SOME. Le condizioni di disuguaglianza non sono supportate.

> **Note**
>
> Il predicato ALL non è supportato.

<> ALL

La parola chiave ALL è sinonima di NOT IN (vedere la condizione Condizione IN) e restituisce true se l'espressione non è compresa nei risultati della query secondaria. Amazon Redshift supporta la condizione <> or != (diverso) solo per ALL. Altre condizioni di confronto non sono supportate.

IS TRUE/FALSE/UNKNOWN

I valori diversi da zero equivalgono a TRUE, 0 equivale a FALSE e null equivale a UNKNOWN. consultare il tipo di dati Tipo booleano.

Esempi

Di seguito sono elencati alcuni semplici esempi di condizioni di confronto:

```
a = 5
a < b
min(x) >= 5
qtysold = any (select qtysold from sales where dateid = 1882
```

La query seguente restituisce sedi con più di 10.000 posti a sedere dalla tabella VENUE:

```
select venueid, venuename, venueseats from venue
where venueseats > 10000
order by venueseats desc;

venueid |           venuename            | venueseats
---------+--------------------------------+------------
83 | FedExField                     |      91704
 6 | New York Giants Stadium        |      80242
79 | Arrowhead Stadium              |      79451
78 | INVESCO Field                  |      76125
69 | Dolphin Stadium                |      74916
67 | Ralph Wilson Stadium           |      73967
76 | Jacksonville Municipal Stadium |      73800
89 | Bank of America Stadium        |      73298
72 | Cleveland Browns Stadium       |      73200
86 | Lambeau Field                  |      72922
...
(57 rows)
```

Questo esempio seleziona dalla tabella USERS gli utenti (USERID) ai quali piace la musica rock:

```
select userid from users where likerock = 't' order by 1 limit 5;

userid
--------
3
5
6
13
16
(5 rows)
```

Questo esempio seleziona dalla tabella USERS gli utenti (USERID) per i quali si sa se gli piace la musica rock:

```
select firstname, lastname, likerock
from users
where likerock is unknown
order by userid limit 10;

firstname | lastname | likerock
----------+----------+----------
Rafael    | Taylor   |
Vladimir  | Humphrey |
Barry     | Roy      |
Tamekah   | Juarez   |
Mufutau   | Watkins  |
Naida     | Calderon |
Anika     | Huff     |
Bruce     | Beck     |
Mallory   | Farrell  |
Scarlett  | Mayer    |
(10 rows
```

Esempi con una colonna TIME

La tabella di esempio seguente TIME_TEST ha una colonna TIME_VAL (tipo TIME) con tre valori inseriti.

```
select time_val from time_test;

time_val
---------------------
20:00:00
00:00:00.5550
00:58:00
```

Nell'esempio seguente vengono estratte le ore da ogni timetz_val.

```
select time_val from time_test where time_val < '3:00';
   time_val
---------------
 00:00:00.5550
 00:58:00
```

L'esempio seguente confronta due valori letterali temporali.

```
select time '18:25:33.123456' = time '18:25:33.123456';
 ?column?
----------
 t
```

Esempi con una colonna TIMETZ

La tabella di esempio seguente TIMETZ_TEST ha una colonna TIMETZ_VAL (tipo TIMETZ) con tre valori inseriti.

```
select timetz_val from timetz_test;

timetz_val
------------------
04:00:00+00
00:00:00.5550+00
05:58:00+00
```

Nell'esempio seguente vengono selezionati solo i valori TIMETZ inferiori a `3:00:00 UTC`. Il confronto viene effettuato dopo aver convertito il valore in UTC.

```
select timetz_val from timetz_test where timetz_val < '3:00:00 UTC';

   timetz_val
---------------
 00:00:00.5550+00
```

L'esempio seguente confronta due valori letterali TIMETZ. Il fuso orario viene ignorato per il confronto.

```
select time '18:25:33.123456 PST' < time '19:25:33.123456 EST';

 ?column?
----------
 t
```

Condizioni logiche

Le condizioni logiche combinano il risultato di due condizioni per produrre un singolo risultato. Tutte le condizioni logiche sono operatori binari con un tipo restituito booleano.

Sintassi

```
expression
{ AND | OR }
expression
NOT expression
```

Le condizioni logiche usano una logica booleana a tre valori in cui il valore null rappresenta una relazione sconosciuta. Nella tabella riportata di seguito sono descritti i risultati delle condizioni logiche, dove E1 ed E2 rappresentano espressioni:

E1	E2	E1 ed E2	E1 o E2	NO E2
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE
TRUE	UNKNOWN	UNKNOWN	TRUE	UNKNOWN
FALSE	TRUE	FALSE	TRUE	
FALSE	FALSE	FALSE	FALSE	
FALSE	UNKNOWN	FALSE	UNKNOWN	
UNKNOWN	TRUE	UNKNOWN	TRUE	
UNKNOWN	FALSE	FALSE	UNKNOWN	
UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	

L'operatore NOT viene valutato prima di AND e l'operatore AND viene valutato prima dell'operatore OR. Eventuali parentesi utilizzate potrebbero sostituire questo ordine di valutazione predefinito.

Esempi

L'esempio seguente restituisce USERID e USERNAME dalla tabella USERS se agli utenti piacciono sia Las Vegas sia gli sport:

```
select userid, username from users
where likevegas = 1 and likesports = 1
order by userid;

userid | username
--------+----------
1 | JSG99FHE
67 | TWU10MZT
87 | DUF19VXU
92 | HYP36WEQ
109 | FPL38HZK
120 | DMJ24GUZ
123 | QZR22XGQ
130 | ZQC82ALK
133 | LBN45WCH
144 | UCX04JKN
165 | TEY68OEB
169 | AYQ83HGO
184 | TVX65AZX
...
(2128 rows)
```

L'esempio successivo restituisce USERID e USERNAME dalla tabella USERS se agli utenti piacciono Las Vegas o gli sport o entrambi. Questa query restituisce tutti gli output dell'esempio precedente più gli utenti a cui piacciono solo Las Vegas o gli sport.

```
select userid, username from users
where likevegas = 1 or likesports = 1
order by userid;

userid | username
--------+----------
1 | JSG99FHE
2 | PGL08LJI
3 | IFT66TXU
5 | AEB55QTM
6 | NDQ15VBM
9 | MSD36KVR
```

```
10 | WKW41AIW
13 | QTF33MCG
15 | OWU78MTR
16 | ZMG93CDD
22 | RHT62AGI
27 | KOY02CVE
29 | HUH27PKK
...
(18968 rows)
```

La query seguente usa le parentesi intorno alla condizione OR per trovare i luoghi a New York o in California in cui è stato rappresentato Macbeth:

```
select distinct venuename, venuecity
from venue join event on venue.venueid=event.venueid
where (venuestate = 'NY' or venuestate = 'CA') and eventname='Macbeth'
order by 2,1;

venuename                |   venuecity
-----------------------------------------+---------------
Geffen Playhouse                         | Los Angeles
Greek Theatre                            | Los Angeles
Royce Hall                               | Los Angeles
American Airlines Theatre                | New York City
August Wilson Theatre                    | New York City
Belasco Theatre                          | New York City
Bernard B. Jacobs Theatre                | New York City
...
```

La rimozione delle parentesi in questo esempio modifica la logica e i risultati della query.

L'esempio seguente utilizza l'operatore NOT:

```
select * from category
where not catid=1
order by 1;

catid | catgroup |  catname  |                 catdesc
-------+----------+-----------+--------------------------------------------
2 | Sports   | NHL       | National Hockey League
3 | Sports   | NFL       | National Football League
4 | Sports   | NBA       | National Basketball Association
5 | Sports   | MLS       | Major League Soccer
```

```
...
```

L'esempio seguente utilizza una condizione NOT seguita da una condizione AND:

```
select * from category
where (not catid=1) and catgroup='Sports'
order by catid;

catid | catgroup | catname |             catdesc
-------+----------+---------+---------------------------------
2 | Sports   | NHL     | National Hockey League
3 | Sports   | NFL     | National Football League
4 | Sports   | NBA     | National Basketball Association
5 | Sports   | MLS     | Major League Soccer
(4 rows)
```

Condizioni di corrispondenza di modelli

Argomenti

- LIKE
- SIMILAR TO
- Operatori POSIX

Un operatore di corrispondenza di modelli cerca una stringa in base a un modello specificato nell'espressione condizionale e restituisce true o false a seconda se trova la corrispondenza. Amazon Redshift utilizza tre metodi per la corrispondenza dei modelli:

- Espressioni LIKE

  L'operatore LIKE confronta un'espressione di stringa, come il nome di colonna, con un modello che usa i caratteri jolly % (percentuale) e _ (sottolineatura). La corrispondenza di modello LIKE copre sempre l'intera stringa. LIKE esegue una corrispondenza che distingue tra maiuscole e minuscole e ILIKE esegue una corrispondenza che non distingue tra maiuscole e minuscole.
- Espressioni regolari SIMILAR TO

  L'operatore SIMILAR TO esegue una corrispondenza tra un'espressione di stringa e un modello di espressione regolare standard SQL, che può comprendere un set di metacaratteri di corrispondenza di modelli che comprende i due supportati dall'operatore LIKE. SIMILAR TO esegue una corrispondenza sull'intera stringa che distingue tra maiuscole e minuscole.

- Espressioni regolari in stile POSIX

  Le espressioni regolari POSIX forniscono un mezzo più potente per la corrispondenza di modelli rispetto agli operatori LIKE e SIMILAR TO. I modelli di espressioni regolari POSIX possono effettuare una corrispondenza su qualsiasi porzione della stringa ed eseguono una corrispondenza che distingue tra maiuscole e minuscole.

La corrispondenza di espressioni regolari, usando operatori SIMILAR TO o POSIX, è costosa in termini di calcolo. Consigliamo di usare LIKE quando possibile, soprattutto se si elaborano grandi quantità di righe. Ad esempio, le query seguenti sono identiche dal punto di vista funzionale, ma la query che usa LIKE viene eseguita molto più velocemente rispetto alla query che usa un'espressione regolare:

```
select count(*) from event where eventname SIMILAR TO '%(Ring|Die)%';
select count(*) from event where eventname LIKE '%Ring%' OR eventname LIKE '%Die%';
```

LIKE

L'operatore LIKE confronta un'espressione di stringa, come il nome di colonna, con un modello che usa i caratteri jolly % (percentuale) e _ (sottolineatura). La corrispondenza di modello LIKE copre sempre l'intera stringa. Per trovare la corrispondenza con una sequenza in qualsiasi parte di una stringa, è necessario che il modello inizi e termini con un segno di percentuale.

LIKE fa distinzione tra maiuscole e minuscole, ILIKE no.

Sintassi

```
expression [ NOT ] LIKE | ILIKE pattern [ ESCAPE 'escape_char' ]
```

Argomenti

espressione

Un'espressione di caratteri UTF-8 valida, come un nome di colonna.

LIKE | ILIKE

LIKE esegue una corrispondenza di modello che fa distinzione tra maiuscole e minuscole. ILIKE esegue una corrispondenza di modello che non fa distinzione tra maiuscole e minuscole per caratteri UTF-8 (ASCII) a byte singolo. Per eseguire una corrispondenza di modello che non

fa distinzione tra maiuscole e minuscole per caratteri multibyte, utilizzare la funzione LOWER sull'espressione e modello con una condizione LIKE.

Al contrario dei predicati di confronto, come = e <>, i predicati LIKE e ILIKE non ignorano implicitamente gli spazi finali. Per ignorare gli spazi finali, utilizzare RTRIM o trasmettere in modo esplicito una colonna CHAR a VARCHAR.

L'operatore ~~ è equivalente a LIKE e ~~* è equivalente a ILIKE. Inoltre, gli operatori !~~ e !~~* sono equivalenti a NOT LIKE e NOT ILIKE.

pattern

Un'espressione di caratteri UTF-8 valida con il modello da associare.

escape_char

Un'espressione di caratteri che eseguirà l'escape dei metacaratteri nel modello. Per impostazione predefinita sono due barre rovesciate ("\\").

Se il modello non contiene metacaratteri, allora il modello rappresenta solo la stringa stessa; in questo caso LIKE agisce come l'operatore di uguaglianza.

Entrambe le espressioni di caratteri possono essere tipi di dati CHAR o VARCHAR. Se differiscono, Amazon Redshift converte il modello al tipo di dati dell'espressione.

LIKE supporta i seguenti metacaratteri di corrispondenza di modelli:

Operatore	Descrizione
%	Abbina qualsiasi sequenza di zero o più caratteri.
_	Abbina qualsiasi carattere singolo.

Esempi

La tabella seguente mostra esempi di corrispondenza di modelli usando LIKE:

Expression	Valori restituiti
`'abc' LIKE 'abc'`	True

Expression	Valori restituiti
`'abc' LIKE 'a%'`	True
`'abc' LIKE '_B_'`	False
`'abc' ILIKE '_B_'`	True
`'abc' LIKE 'c%'`	False

L'esempio seguente trova tutte le città il cui nome inizia per "E":

```
select distinct city from users
where city like 'E%' order by city;
city
---------------
East Hartford
East Lansing
East Rutherford
East St. Louis
Easthampton
Easton
Eatontown
Eau Claire
...
```

L'esempio seguente trova tutti gli utenti il cui cognome contiene "ten":

```
select distinct lastname from users
where lastname like '%ten%' order by lastname;
lastname
-------------
Christensen
Wooten
...
```

Nell'esempio seguente viene illustrato come mettere in corrispondenza più modelli.

```
select distinct lastname from tickit.users
where lastname like 'Chris%' or lastname like '%Wooten' order by lastname;
lastname
```

```
-------------
Christensen
Christian
Wooten
...
```

L'esempio seguente trova le città il cui terzo e quarto carattere sono "ea". Il comando usa ILIKE per dimostrare che non fa distinzione tra maiuscole e minuscole:

```
select distinct city from users where city ilike '__EA%' order by city;
city
-------------
Brea
Clearwater
Great Falls
Ocean City
Olean
Wheaton
(6 rows)
```

Nell'esempio seguente viene usata la stringa con caratteri escape predefinita (\\) per ricercare stringhe che contengono "start_" (il testo `start` seguito da una sottolineatura _):

```
select tablename, "column" from pg_table_def
where "column" like '%start\\_%'
limit 5;

     tablename     |    column
-------------------+---------------
 stl_s3client      | start_time
 stl_tr_conflict   | xact_start_ts
 stl_undone        | undo_start_ts
 stl_unload_log    | start_time
 stl_vacuum_detail | start_row
(5 rows)
```

L'esempio seguente specifica "^" come carattere di escape, quindi usa il carattere di escape per ricercare stringhe che contengono "start_" (il testo `start` seguito da una sottolineatura _):

```
select tablename, "column" from pg_table_def
where "column" like '%start^_%' escape '^'
limit 5;
```

```
     tablename     |    column
-------------------+---------------
 stl_s3client      | start_time
 stl_tr_conflict   | xact_start_ts
 stl_undone        | undo_start_ts
 stl_unload_log    | start_time
 stl_vacuum_detail | start_row
(5 rows)
```

L'esempio seguente utilizza l'operatore ~~* per eseguire una ricerca senza distinzione tra maiuscole e minuscole (ILIKE) delle città che iniziano con "Ag".

```
select distinct city from users where city ~~* 'Ag%' order by city;

city
------------
Agat
Agawam
Agoura Hills
Aguadilla
```

SIMILAR TO

L'operatore SIMILAR TO associa un'espressione di stringa, come il nome di colonna, a un modello di espressione regolare standard SQL. Un modello di espressione regolare SQL può comprendere un set di metacaratteri di corrispondenza di modelli, compresi i due supportati dall'operatore LIKE LIKE.

L'operatore SIMILAR TO restituisce true solo se il modello corrisponde all'intera stringa, a differenza del comportamento dell'espressione regolare POSIX, in cui il modello può corrispondere a qualsiasi porzione della stringa.

SIMILAR TO esegue una corrispondenza che fa distinzione tra maiuscole e minuscole.

> **Note**
>
> La corrispondenza di espressioni regolari usando SIMILAR TO è costosa in termini di calcolo. Consigliamo di usare LIKE quando possibile, soprattutto se si elaborano grandi quantità di righe. Ad esempio, le query seguenti sono identiche dal punto di vista funzionale, ma la query che usa LIKE viene eseguita molto più velocemente rispetto alla query che usa un'espressione regolare:

```
select count(*) from event where eventname SIMILAR TO '%(Ring|Die)%';
select count(*) from event where eventname LIKE '%Ring%' OR eventname LIKE '%Die
%';
```

Sintassi

```
expression [ NOT ] SIMILAR TO pattern [ ESCAPE 'escape_char' ]
```

Argomenti

espressione

Un'espressione di caratteri UTF-8 valida, come un nome di colonna.

SIMILAR TO

SIMILAR TO esegue una corrispondenza di modello che distingue tra maiuscole e minuscole per l'intera stringa nell'espressione.

pattern

Un'espressione di caratteri UTF-8 valida che rappresenta un modello di espressione regolare standard SQL.

escape_char

Un'espressione di caratteri che eseguirà l'escape dei metacaratteri nel modello. Per impostazione predefinita sono due barre rovesciate ("\\").

Se il modello non contiene metacaratteri, allora il modello rappresenta solo la stringa stessa.

Entrambe le espressioni di caratteri possono essere tipi di dati CHAR o VARCHAR. Se differiscono, Amazon Redshift converte il modello al tipo di dati dell'espressione.

SIMILAR TO supporta i seguenti metacaratteri di corrispondenza di modelli:

Operatore	Descrizione
%	Abbina qualsiasi sequenza di zero o più caratteri.

Operatore	Descrizione
_	Abbina qualsiasi carattere singolo.
|	Denota alternanza (una di due alternative).
*	Ripeti la voce precedente zero o più volte.
+	Ripeti la voce precedente una o più volte.
?	Ripeti la voce precedente zero o una volta.
{m}	Ripetere la voce precedente esattamente m volte.
{m,}	Ripetere la voce precedente m o più volte.
{m,n}	Ripetere la voce precedente almeno m volte e non più di n volte.
()	Raggruppa tra parentesi voci di gruppo in una singola voce logica.
[...]	Un'espressione tra parentesi specifica una classe di caratteri, come nelle espressioni regolari POSIX.

Esempi

La tabella riportata di seguito mostra esempi di corrispondenza di modelli usando SIMILAR TO:

Expression	Valori restituiti
`'abc' SIMILAR TO 'abc'`	True
`'abc' SIMILAR TO '_b_'`	True
`'abc' SIMILAR TO '_A_'`	False
`'abc' SIMILAR TO '%(b|d)%'`	True
`'abc' SIMILAR TO '(b|c)%'`	False

Expression	Valori restituiti
`'AbcAbcdefgefg12efgefg12' SIMILAR TO '((Ab)?c)+d((efg)+(12))+'`	True
`'aaaaaab11111xy' SIMILAR TO 'a{6}_[0-9]{5}(x|y){2}'`	True
`'$0.87' SIMILAR TO '$[0-9]+(.[0-9][0-9])?'`	True

L'esempio seguente trova tutte le città il cui nome contiene "E" o "H":

```
SELECT DISTINCT city FROM users
WHERE city SIMILAR TO '%E%|%H%' ORDER BY city LIMIT 5;

      city
-----------------
 Agoura Hills
 Auburn Hills
 Benton Harbor
 Beverly Hills
 Chicago Heights
```

Nell'esempio seguente viene usata la stringa di escape predefinita ("\\") per cercare stringhe che contengono "_":

```
SELECT tablename, "column" FROM pg_table_def
WHERE "column" SIMILAR TO '%start\\_%'
ORDER BY tablename, "column" LIMIT 5;

        tablename         |       column
--------------------------+---------------------
 stcs_abort_idle          | idle_start_time
 stcs_abort_idle          | txn_start_time
 stcs_analyze_compression | start_time
 stcs_auto_worker_levels  | start_level
 stcs_auto_worker_levels  | start_wlm_occupancy
```

L'esempio seguente specifica "^" come stringa di escape, quindi usa la stringa di escape per cercare stringhe che contengono "_":

```
SELECT tablename, "column" FROM pg_table_def
WHERE "column" SIMILAR TO '%start^_%' ESCAPE '^'
ORDER BY tablename, "column" LIMIT 5;

        tablename         |       column
--------------------------+--------------------
 stcs_abort_idle          | idle_start_time
 stcs_abort_idle          | txn_start_time
 stcs_analyze_compression | start_time
 stcs_auto_worker_levels  | start_level
 stcs_auto_worker_levels  | start_wlm_occupancy
```

Operatori POSIX

Un'espressione regolare POSIX è una sequenza di caratteri che specifica uno schema di corrispondenza. Una stringa corrisponde a un'espressione regolare se è un membro del set regolare descritto dall'espressione regolare.

Le espressioni regolari POSIX forniscono un mezzo più potente per la corrispondenza di modelli rispetto agli operatori LIKE e SIMILAR TO. I modelli di espressioni regolari POSIX possono corrispondere a qualsiasi porzione di una stringa, a differenza dell'operatore SIMILAR TO che restituisce true solo se il modello corrisponde all'intera stringa.

> **Note**
>
> La corrispondenza di espressioni regolari usando operatori POSIX è costosa in termini di calcolo. Consigliamo di usare LIKE quando possibile, soprattutto se si elaborano grandi quantità di righe. Ad esempio, le query seguenti sono identiche dal punto di vista funzionale, ma la query che usa LIKE viene eseguita molto più velocemente rispetto alla query che usa un'espressione regolare:
>
> ```
> select count(*) from event where eventname ~ '.*(Ring|Die).*';
> select count(*) from event where eventname LIKE '%Ring%' OR eventname LIKE '%Die
> %';
> ```

Sintassi

```
expression [ ! ] ~ pattern
```

Argomenti

espressione

Un'espressione di caratteri UTF-8 valida, come un nome di colonna.

!

Operatore di negazione. Non corrisponde all'espressione regolare.

~

Eseguire una corrispondenza che distingue tra maiuscole e minuscole su qualsiasi sottostringa dell'espressione.

> **Note**
>
> Un ~~ è sinonimo di LIKE.

pattern

Un valore letterale di stringa che rappresenta un modello di espressione regolare.

Se il modello non contiene caratteri jolly, allora il modello rappresenta solo la stringa stessa.

Per cercare stringhe che comprendono metacaratteri, come ". * | ? ", ecc., eseguire l'escape del carattere usando due barre rovesciate (" \\"). A differenza di SIMILAR TO e LIKE, la sintassi dell'espressione regolare POSIX non supporta un carattere di escape definito dall'utente.

Entrambe le espressioni di caratteri possono essere tipi di dati CHAR o VARCHAR. Se differiscono, Amazon Redshift converte il modello al tipo di dati dell'espressione.

Tutte le espressioni di caratteri possono essere tipi di dati CHAR o VARCHAR. Se l'espressione differisce per tipo di dati, Amazon Redshift la converte nel tipo di dati dell'espressione.

La corrispondenza di modelli POSIX supporta i seguenti metacaratteri:

POSIX	Descrizione
.	Abbina qualsiasi carattere singolo.
*	Associa zero o più occorrenze.
+	Associa una o più occorrenze.
?	Associa zero o una occorrenza.
|	Specifica corrispondenze alternative; ad esempio, `E | H` significa E o H.
^	Corrisponde al personaggio. beginning-of-line
$	Corrisponde al end-of-line personaggio.
$	Associa la fine della riga.
[]	Le parentesi specificano un elenco di corrispondenza, che deve corrispon dere a un'espressione nell'elenco. Un accento circonflesso (^) precede un elenco di non corrispondenze, che associa qualsiasi carattere ad eccezione delle espressioni rappresentate nell'elenco.
( )	Raggruppa tra parentesi voci di gruppo in una singola voce logica.
{m}	Ripetere la voce precedente esattamente m volte.
{m,}	Ripetere la voce precedente m o più volte.
{m,n}	Ripetere la voce precedente almeno m volte e non più di n volte.
[: :]	Associa qualsiasi carattere in una classe di caratteri POSIX. Nelle classi di caratteri seguenti, Amazon Redshift supporta solo i caratteri ASCII `[:alnum:]` , `[:alpha:]` , `[:lower:]` e `[:upper:]`

Amazon Redshift supporta le seguenti classi di caratteri POSIX.

Classe di caratteri	Descrizione
`[[:alnum:]]`	Tutti i caratteri alfanumerici ASCII
`[[:alpha:]]`	Tutti i caratteri alfabetici ASCII
`[[:blank:]]`	Tutti i caratteri di spazio vuoto
`[[:cntrl:]]`	Tutti i caratteri di controllo (non stampabili)
`[[:digit:]]`	Tutte le cifre numeriche
`[[:lower:]]`	Tutti i caratteri alfabetici ASCII minuscoli
`[[:punct:]]`	Tutti i caratteri di punteggiatura
`[[:space:]]`	Tutti i caratteri di spazio (non stampabili)
`[[:upper:]]`	Tutti i caratteri alfabetici ASCII maiuscoli
`[[:xdigit:]]`	Tutti i caratteri esadecimali validi

Amazon Redshift supporta i seguenti operatori influenzati da Perl in espressioni regolari. Esegui l'escape dell'operatore usando due barre rovesciate (‘\\').

Operatore	Descrizione	Espressione di classe di caratteri equivalente
`\\d`	Un carattere cifra	`[[:digit:]]`
`\\D`	Un carattere non cifra	`[^[:digit:]]`
`\\w`	Un carattere parola	`[[:word:]]`
`\\W`	Un carattere non parola	`[^[:word:]]`
`\\s`	Un carattere di spazio vuoto	`[[:space:]]`
`\\S`	Un carattere di spazio non vuoto	`[^[:space:]]`

Operatore	Descrizione	Espressione di classe di caratteri equivalente
\\b	Una parola limite	

Esempi

La tabella riportata di seguito mostra esempi di corrispondenza di modelli usando operatori POSIX:

Expression	Valori restituiti
`'abc' ~ 'abc'`	True
`'abc' ~ 'a'`	True
`'abc' ~ 'A'`	False
`'abc' ~ '.*(b|d).*'`	True
`'abc' ~ '(b|c).*'`	True
`'AbcAbcdefgefg12efgefg12' ~ '((Ab)?c)+d((efg)+(12))+'`	True
`'aaaaaab11111xy' ~ 'a{6}.[1]{5}(x|y){2}'`	True
`'$0.87' ~ '\\$[0-9]+(\\.[0-9][0-9])?'`	True
`'ab c' ~ '[[:space:]]'`	True
`'ab c' ~ '\\s'`	True
`' ' ~ '\\S'`	False

L'esempio seguente trova tutte le città il cui nome contiene E o H:

```
SELECT DISTINCT city FROM users
```

```
WHERE city ~ '.*E.*|.*H.*' ORDER BY city LIMIT 5;

      city
-----------------
 Agoura Hills
 Auburn Hills
 Benton Harbor
 Beverly Hills
 Chicago Heights
```

L'esempio seguente trova le città il cui nome non contiene E o H:

```
SELECT DISTINCT city FROM users WHERE city !~ '.*E.*|.*H.*' ORDER BY city LIMIT 5;

      city
-----------------
 Aberdeen
 Abilene
 Ada
 Agat
 Agawam
```

Nell'esempio seguente viene usata la stringa di escape predefinita ("\\") per cercare stringhe che contengono un punto.

```
SELECT venuename FROM venue
WHERE venuename ~ '.*\\..*'
ORDER BY venueid;

          venuename
------------------------------
 St. Pete Times Forum
 Jobing.com Arena
 Hubert H. Humphrey Metrodome
 U.S. Cellular Field
 Superpages.com Center
 E.J. Nutter Center
 Bernard B. Jacobs Theatre
 St. James Theatre
```

Condizione di intervallo BETWEEN

Una condizione BETWEEN testa le espressioni per l'inclusione in un intervallo di valori, usando le parole chiave BETWEEN e AND.

Sintassi

```
expression [ NOT ] BETWEEN expression AND expression
```

Le espressioni possono essere numeriche, di caratteri o datetime, ma è necessario che siano compatibili. L'intervallo è inclusivo.

Esempi

Il primo esempio conta quante transazioni hanno registrato vendite di 2, 3 o 4 biglietti:

```
select count(*) from sales
where qtysold between 2 and 4;

count
--------
104021
(1 row)
```

La condizione di intervallo comprende i valori di inizio e di fine.

```
select min(dateid), max(dateid) from sales
where dateid between 1900 and 1910;

min  | max
-----+-----
1900 | 1910
```

È necessario che la prima espressione in una condizione di intervallo sia il valore minore e la seconda espressione il valore maggiore. L'esempio seguente restituirà sempre zero righe a causa dei valori delle espressioni:

```
select count(*) from sales
where qtysold between 4 and 2;
```

```
count
-------
0
(1 row)
```

Tuttavia, l'applicazione del modificatore NOT invertirà la logica e produrrà un conto di tutte le righe:

```
select count(*) from sales
where qtysold not between 4 and 2;

count
--------
172456
(1 row)
```

La query seguente restituisce un elenco di sedi con 20.000-50.000 posti a sedere:

```
select venueid, venuename, venueseats from venue
where venueseats between 20000 and 50000
order by venueseats desc;

venueid |       venuename               | venueseats
---------+-------------------------------+------------
116 | Busch Stadium                 |      49660
106 | Rangers BallPark in Arlington |      49115
96 | Oriole Park at Camden Yards   |      48876
...
(22 rows)
```

L'esempio seguente mostra l'utilizzo di BETWEEN per i valori di data:

```
select salesid, qtysold, pricepaid, commission, saletime
from sales
where eventid between 1000 and 2000
   and saletime between '2008-01-01' and '2008-01-03'
order by saletime asc;

salesid | qtysold | pricepaid | commission |   saletime
--------+---------+-----------+------------+---------------
  65082 |       4 |       472 |       70.8 | 1/1/2008 06:06
 110917 |       1 |       337 |      50.55 | 1/1/2008 07:05
```

```
 112103 |       1 |       241 |      36.15 | 1/2/2008 03:15
 137882 |       3 |      1473 |     220.95 | 1/2/2008 05:18
  40331 |       2 |        58 |        8.7 | 1/2/2008 05:57
 110918 |       3 |      1011 |     151.65 | 1/2/2008 07:17
  96274 |       1 |       104 |       15.6 | 1/2/2008 07:18
 150499 |       3 |       135 |      20.25 | 1/2/2008 07:20
  68413 |       2 |       158 |       23.7 | 1/2/2008 08:12
```

È importante notare che, sebbene l'intervallo di BETWEEN sia inclusivo, le date hanno un valore di ora predefinito di 00:00:00. L'unica riga valida del 3 gennaio per la query di esempio sarebbe una riga con saletime `1/3/2008 00:00:00`.

Condizione Null

La condizione null testa i valori null, quando un valore manca o è sconosciuto.

Sintassi

```
expression IS [ NOT ] NULL
```

Argomenti

espressione

Qualsiasi espressione come una colonna.

IS NULL

È true quando il valore dell'espressione è null e false quando ha un valore.

IS NOT NULL

È false quando il valore dell'espressione è null e true quando ha un valore.

Esempio

Questo esempio indica quante volte la tabella SALES contiene un valore null nel campo QTYSOLD:

```
select count(*) from sales
where qtysold is null;
count
```

```
-------
0
(1 row)
```

Condizione EXISTS

Le condizioni EXISTS testano l'esistenza di righe in una subquery e restituiscono true se una subquery restituisce almeno una riga. Se viene specificato NOT, la condizione restituisce true se una subquery restituisce nessuna riga.

Sintassi

```
[ NOT ] EXISTS (table_subquery)
```

Argomenti

EXISTS

È true se table_subquery restituisce almeno una riga.

NOT EXISTS

È true se table_subquery restituisce nessuna riga.

table_subquery

Una subquery che viene valutata una tabella con una o più colonne e una o più righe.

Esempio

Questo esempio restituisce tutti gli identificatori di data, una volta ciascuno, per ogni data che ha registrato una vendita di qualsiasi tipo:

```
select dateid from date
where exists (
select 1 from sales
where date.dateid = sales.dateid
)
order by dateid;

dateid
```

```
--------
1827
1828
1829
...
```

Condizione IN

Una condizione IN verifica se un valore appartiene a un set di valori o a una subquery.

Sintassi

```
expression [ NOT ] IN (expr_list | table_subquery)
```

Argomenti

espressione

Un'espressione datetime, di caratteri o numerica che viene valutata rispetto a expr_list o table_subquery e deve essere compatibile con il tipo di dati di quell'elenco o subquery.

expr_list

Una o più espressioni delimitate da virgola o uno o più set di espressioni delimitate da virgola racchiusi tra parentesi.

table_subquery

Una subquery che viene valutata una tabella con una o più righe ma che è limitata a una sola colonna nel suo elenco di selezione.

IN | NOT IN

IN restituisce true se l'espressione è un membro della query o dell'elenco di espressioni. NOT IN restituisce true se l'espressione non è un membro. IN e NOT IN restituiscono NULL e nessuna riga nei casi seguenti: se l'espressione genera un valore null; se non ci sono valori expr_list o table_subquery corrispondenti e almeno una di queste righe di confronto restituisce un valore null.

Esempi

Le condizioni seguenti sono true solo per quei valori elencati:

```
qtysold in (2, 4, 5)
date.day in ('Mon', 'Tues')
date.month not in ('Oct', 'Nov', 'Dec')
```

Ottimizzazione per grandi elenchi IN

Per ottimizzare l'esecuzione delle query, un elenco IN che comprende più di 10 valori viene internamente valutato come un array scalare. Gli elenchi IN con meno di 10 valori vengono valutati come una serie di predicati OR. Questa ottimizzazione è supportata per i tipi di dati SMALLINT, INTEGER, BIGINT, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP e TIMESTAMPTZ.

Guarda l'output di EXPLAIN per la query per vedere l'effetto di questa ottimizzazione. Ad esempio:

```
explain select * from sales
QUERY PLAN
------------------------------------------------------------------------
XN Seq Scan on sales  (cost=0.00..6035.96 rows=86228 width=53)
Filter: (salesid = ANY ('{1,2,3,4,5,6,7,8,9,10,11}'::integer[]))
(2 rows)
```

Comandi SQL

Il linguaggio SQL è composto da comandi che usi per creare e manipolare oggetti di database, eseguire query, caricare tabelle e modificare i dati nelle tabelle.

Amazon Redshift è basato su PostgreSQL. Amazon Redshift e PostgreSQL si differenziano per un certo numero di aspetti molto importanti, di cui bisogna tener conto mentre si progettano e sviluppano le applicazioni di data warehouse. Per informazioni sulle differenze tra Amazon Redshift SQL e PostgreSQL, consultare Amazon Redshift e PostgreSQL.

> **Note**
>
> Le dimensioni massime per una istruzione SQL è di 16 MB.

Argomenti

- ABORT

- ALTER DATABASE
- ALTER DATASHARE
- ALTER DEFAULT PRIVILEGES
- ALTER EXTERNAL VIEW (anteprima)
- ALTER FUNCTION
- ALTER GROUP
- ALTER IDENTITY PROVIDER
- ALTER MASKING POLICY
- ALTER MATERIALIZED VIEW
- ALTER RLS POLICY
- ALTER ROLE
- ALTER PROCEDURE
- ALTER SCHEMA
- ALTER SYSTEM
- ALTER TABLE
- ALTER TABLE APPEND
- ALTER USER
- ANALYZE
- ANALYZE COMPRESSION
- ATTACH MASKING POLICY
- ATTACH RLS POLICY
- BEGIN
- CALL
- CANCEL
- CLOSE
- COMMENT
- COMMIT
- COPY

- CREATE DATABASE
- CREARE DATASHARE
- CREATE EXTERNAL FUNCTION
- CREATE EXTERNAL SCHEMA
- CREATE EXTERNAL TABLE
- CREATE EXTERNAL VIEW (anteprima)
- CREATE FUNCTION
- CREATE GROUP
- CREATE IDENTITY PROVIDER
- CREATE LIBRARY
- CREATE MASKING POLICY
- CREATE MATERIALIZED VIEW
- CREATE MODEL
- CREATE PROCEDURE
- CREATE RLS POLICY
- CREATE ROLE
- CREATE SCHEMA
- CREATE TABLE
- CREATE TABLE AS
- CREA UTENTE
- CREATE VIEW
- DEALLOCATE
- DECLARE
- DELETE
- DESC DATASHARE
- DESC IDENTITY PROVIDER
- DETACH MASKING POLICY
- DETACH RLS POLICY

- DROP DATABASE
- DROP DATASHARE
- DROP EXTERNAL VIEW (anteprima)
- DROP FUNCTION
- DROP GROUP
- DROP IDENTITY PROVIDER
- DROP LIBRARY
- DROP MASKING POLICY
- DROP MODEL
- DROP MATERIALIZED VIEW
- DROP PROCEDURE
- DROP RLS POLICY
- DROP ROLE
- DROP SCHEMA
- DROP TABLE
- DROP USER
- DROP VIEW
- END
- EXECUTE
- EXPLAIN
- FETCH
- GRANT
- INSERT
- INSERT (tabella esterna)
- LOCK
- MERGE
- PREPARE
- REFRESH MATERIALIZED VIEW

- RESET
- REVOKE
- ROLLBACK
- SELECT
- SELECT INTO
- SET
- SET SESSION AUTHORIZATION
- SET SESSION CHARACTERISTICS
- MOSTRA
- SHOW COLUMNS
- SHOW EXTERNAL TABLE
- SHOW DATABASES
- SHOW MODEL
- SHOW DATASHARES
- SHOW PROCEDURE
- SHOW SCHEMAS
- SHOW TABLE
- SHOW TABLES
- SHOW VIEW
- START TRANSACTION
- TRUNCATE
- UNLOAD
- UPDATE
- VACUUM

ABORT

Interrompe la transazione attualmente in esecuzione ed elimina tutti gli aggiornamenti apportati da quella transazione. ABORT non ha alcun effetto sulle transazioni già completate.

Questo comando esegue la stessa funzione del comando ROLLBACK. Per informazioni, consultare ROLLBACK.

Sintassi

```
ABORT [ WORK | TRANSACTION ]
```

Parametri

WORK

Parola chiave facoltativa.

TRANSACTION

Parola chiave facoltativa; WORK e TRANSACTION sono sinonimi.

Esempio

L'esempio seguente crea una tabella quindi avvia una transazione in cui i dati vengono inseriti nella tabella. Il comando ABORT esegue quindi il rollback dell'inserimento dei dati per lasciare vuota la tabella.

Il seguente comando crea una tabella di esempio chiamata MOVIE_GROSS:

```
create table movie_gross( name varchar(30), gross bigint );
```

Il prossimo set di comandi avvia una transazione che inserisce due righe di dati nella tabella:

```
begin;

insert into movie_gross values ( 'Raiders of the Lost Ark', 23400000);

insert into movie_gross values ( 'Star Wars', 10000000 );
```

Successivamente, il seguente comando seleziona i dati dalla tabella per mostrare che è stato inserito:

```
select * from movie_gross;
```

L'output del comando mostra che entrambe le righe sono state inserite:

```
         name           |  gross
------------------------+----------
Raiders of the Lost Ark | 23400000
Star Wars               | 10000000
(2 rows)
```

Questo comando esegue ora il rollback delle modifiche dei dati dove è iniziata la transazione:

```
abort;
```

Selezionando i dati dalla tabella ora mostra una tabella vuota:

```
select * from movie_gross;

 name | gross
------+-------
(0 rows)
```

ALTER DATABASE

Cambia gli attributi di un database.

Privilegi richiesti

Per utilizzare ALTER DATABASE, è richiesto uno dei seguenti privilegi.

- Superuser
- Utenti con il privilegio ALTER DATABASE
- Proprietario del database

Sintassi

```
ALTER DATABASE database_name
{ RENAME TO new_name
| OWNER TO new_owner
| CONNECTION LIMIT { limit | UNLIMITED }
```

```
| COLLATE { CASE_SENSITIVE | CASE_INSENSITIVE }
| ISOLATION LEVEL { SERIALIZABLE | SNAPSHOT }
| INTEGRATION REFRESH {{ ALL | INERROR } TABLES [IN SCHEMA schema [, ...]] |
 TABLE schema.table [, ...]}
}
```

Parametri

database_name

Nome del database da modificare. In genere, si modifica un database a cui non si è attualmente connessi; in ogni caso, le modifiche hanno effetto solo nelle sessioni successive. Puoi modificare il proprietario del database corrente, ma non è possibile rinominarlo:

```
alter database tickit rename to newtickit;
ERROR:  current database may not be renamed
```

RENAME TO

Rinomina il database specificato. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori. Non puoi rinominare i database dev, padb_harvest, template0, template1 o sys:internal e non puoi rinominare il database corrente. Solo il proprietario del database o un superuser (p. 891) può rinominare un database; i proprietari che non sono utenti con privilegi avanzati devono inoltre avere il privilegio CREATEDB.

new_name

Nuovo nome del database.

OWNER TO

Cambia il proprietario del database specificato. Puoi modificare il proprietario del database corrente o di un altro database. Solo un utente con privilegi avanzati può cambiare il proprietario.

new_owner

Nuovo proprietario del database. Il nuovo proprietario deve essere un utente di database esistente con privilegi di scrittura. Per ulteriori informazioni sui privilegi degli utenti, consultare GRANT.

CONNECTION LIMIT { limit | UNLIMITED }

Numero massimo di connessioni di database che gli utenti possono aprire contemporaneamente. Il limite non viene applicato per gli utenti con privilegi avanzati. Utilizza la parola chiave

UNLIMITED per consentire il numero massimo di connessioni simultanee. È possibile che venga applicato anche un limite al numero di connessioni per ciascun utente. Per ulteriori informazioni, consultare CREA UTENTE. Il valore predefinito è UNLIMITED. Per visualizzare le connessioni correnti, eseguire una query sulla vista di sistema STV_SESSIONS.

> **Note**
>
> Se si applicano entrambi i limiti di connessione utente e database, deve essere disponibile uno slot di connessione inutilizzato che rientra in entrambi i limiti quando un utente tenta di connettersi.

COLLATE { CASE_SENSITIVE | CASE_INSENSITIVE }

Una clausola che specifica se la ricerca o il confronto tra stringhe fa distinzione tra maiuscole e minuscole o meno.

È possibile modificare la distinzione tra maiuscole e minuscole del database corrente che è vuoto.

Per modificare la distinzione tra maiuscole e minuscole, è necessario disporre del privilegio sul database corrente. Gli utenti con privilegi avanzati o i proprietari di database con il privilegio CREATE DATABASE possono anche modificare la distinzione tra maiuscole e minuscole.

ISOLATION LEVEL { SERIALIZABLE | SNAPSHOT }

Una clausola che specifica il livello di isolamento utilizzato quando vengono eseguite query su un database.

- Isolamento SERIALIZABLE: fornisce la serializzabilità completa per le transazioni simultanee. Per ulteriori informazioni, consulta Isolamento serializzabile.
- Isolamento SNAPSHOT: fornisce un livello di isolamento con protezione contro i conflitti di aggiornamento ed eliminazione.

Per ulteriori informazioni sui livelli di isolamento, consulta CREATE DATABASE.

Considerare quanto segue durante la modifica del livello di isolamento di un database:

- Per modificare il livello di isolamento del database, è necessario disporre del privilegio utente con privilegi avanzati o del privilegio CREATE DATABASE per il database corrente.
- Non è possibile modificare il livello di isolamento del database `dev`.

- Non è possibile modificare il livello di isolamento all'interno di un blocco di transazioni.
- Il comando alter isolation level non riesce se altri utenti sono connessi al database.
- Il comando alter isolation level può modificare le impostazioni del livello di isolamento della sessione corrente.

INTEGRATION REFRESH {{ ALL | INERROR } TABLES [IN SCHEMA schema [, ...]] | TABLE schema.table [, ...]}

Una clausola che specifica se Amazon Redshift aggiorna tutte le tabelle o le tabelle con errori nello schema o nella tabella specificati. L'aggiornamento attiverà la replica completa delle tabelle nello schema o nella tabella specificati dal database di origine.

Per ulteriori informazioni, consulta Utilizzo delle integrazioni Zero-ETL nella Guida alla gestione di Amazon Redshift. Per ulteriori informazioni sugli stati di integrazione, consulta SVV_INTEGRATION_TABLE_STATE e SVV_INTEGRATION.

Note per l'utilizzo

I comandi di ALTER DATABASE si applicano alle sessioni successive e non alle sessioni correnti. Devi riconnetterti al database modificato per vedere l'effetto della modifica.

Esempi

L'esempio seguente rinomina un database denominato TICKIT_SANDBOX in TICKIT_TEST:

```
alter database tickit_sandbox rename to tickit_test;
```

L'esempio seguente modifica il proprietario del database TICKIT (il database corrente) in DWUSER:

```
alter database tickit owner to dwuser;
```

L'esempio seguente modifica la distinzione tra maiuscole e minuscole del database sampledb:

```
ALTER DATABASE sampledb COLLATE CASE_INSENSITIVE;
```

Nell'esempio seguente viene modificato un database denominato **`sampledb`** con livello di isolamento SNAPSHOT.

```
ALTER DATABASE sampledb ISOLATION LEVEL SNAPSHOT;
```

L'esempio seguente aggiorna le tabelle **`sample_table1`** e **`sample_table2`** del database **`sample_integration_db`** nell'integrazione Zero-ETL.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH TABLES sample_table1,
 sample_table2;
```

L'esempio seguente aggiorna tutte le tabelle sincronizzate e non riuscite nell'integrazione Zero-ETL.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL tables;
```

L'esempio seguente aggiorna tutte le tabelle con `ErrorState` presenti nello schema **`sample_schema`**.

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH INERROR TABLES in SCHEMA
 sample_schema;
```

ALTER DATASHARE

Cambia la definizione di una unità di condivisione dati. È possibile aggiungere o rimuovere oggetti utilizzando ALTER DATASHARE. È possibile modificare una unità di condivisione dati solo nel database corrente. Aggiunta o rimozione di oggetti dal database associato a una unità di condivisione dati. Il proprietario dell'unità di condivisione dati con le autorizzazioni necessarie per gli oggetti dell'unità di condivisione dati da aggiungere o rimuovere può modificare l'unità.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per ALTER DATASHARE:

- Utente con privilegi avanzati.
- Utente con il privilegio ALTER DATASHARE.
- Utenti con privilegio ALTER o ALL sull'unità di condivisione dati.
- Per aggiungere oggetti specifici a una unità di condivisione dati, gli utenti devono avere il privilegio sugli oggetti. Per questo caso, gli utenti devono essere i proprietari degli oggetti o disporre dei privilegi SELECT, USAGE o ALL sugli oggetti.

Sintassi

La sintassi seguente illustra come aggiungere o rimuovere oggetti nell'unità unità di condivisione dati.

```
ALTER DATASHARE datashare_name { ADD | REMOVE } {
TABLE schema.table [, ...]
| SCHEMA schema [, ...]
| FUNCTION schema.sql_udf (argtype,...) [, ...]
| ALL TABLES IN SCHEMA schema [, ...]
| ALL FUNCTIONS IN SCHEMA schema [, ...] }
```

La sintassi seguente illustra come configurare le proprietà dell'unità unità di condivisione dati.

```
ALTER DATASHARE datashare_name {
[ SET PUBLICACCESSIBLE [=] TRUE | FALSE ]
[ SET INCLUDENEW [=] TRUE | FALSE FOR SCHEMA schema ] }
```

Parametri

datashare_name

Il nome della unità di condivisione dati da modificare.

ADD | REMOVE

Una clausola che specifica se aggiungere o rimuovere oggetti dall'unità di condivisione dati.

TABLE schema.table [, ...]

Il nome della tabella o della vista nello schema specificato da aggiungere all'unità di condivisione dati.

SCHEMA schema [, ...]

Il nome dello schema da aggiungere all'unità di condivisione dati.

FUNZIONE schema.sql_udf (argtype,...) [, ...]

Il nome della funzione SQL definita dall'utente da aggiungere all'unità di condivisione dati.

ALL TABLES IN SCHEMA schema [, ...]

Una clausola che specifica se aggiungere tutte le tabelle e le viste nello schema specificato all'unità di condivisione dati.

ALL FUNCTIONS IN SCHEMA schema [, ...] }

Una clausola che specifica l'aggiunta di tutte le funzioni dello schema specificato all'unità di condivisione dati.

[SET PUBLICACCESSIBLE [=] TRUE | FALSE]

Una clausola che specifica se una unità di condivisione dati può essere condivisa in un cluster accessibile pubblicamente.

[SET INCLUDENEW [=] TRUE | FALSE FOR SCHEMA schema]

Una clausola che specifica se aggiungere all'unità di condivisione dati tabelle, viste o funzioni definite dall'utente (FDU) SQL future create nell'unità di condivisione dati specificata. Le tabelle correnti, le viste o le funzioni definite dall'utente SQL nello schema specificato non vengono aggiunte all'unità di condivisione dati. Solo gli utenti con privilegi avanzati possono modificare questa proprietà per ogni coppia unità di condivisione dati-schema. Per impostazione predefinita, la clausola INCLUDENEW è false.

Note per l'utilizzo di ALTER DATASHARE

- I seguenti utenti possono modificare una unità di condivisione dati:
  - Un utente con privilegi avanzati
  - Il proprietario dell'unità di condivisione dati
  - Utenti con privilegio ALTER o ALL sull'unità di condivisione dati
- Per aggiungere oggetti specifici a una unità di condivisione dati, gli utenti devono avere il privilegio sugli oggetti. Gli utenti devono essere i proprietari degli oggetti o disporre dei privilegi SELECT, USAGE o ALL sugli oggetti.
- È possibile condividere schemi, tabelle, viste regolari, viste di associazione tardiva, viste materializzate e funzioni definite dall'utente (FDU) SQL. Aggiungere uno schema all'unità di condivisione dati prima di aggiungere altri oggetti nello schema.

  Quando si aggiunge uno schema, Amazon Redshift non aggiunge tutti gli oggetti sotto di esso. È necessario aggiungerli esplicitamente.
- Ti consigliamo di creare condivisioni di AWS Data Exchange dati con l'impostazione accessibile al pubblico attivata.
- In generale, si consiglia di non modificare un AWS Data Exchange datashare per disattivare l'accessibilità pubblica utilizzando l'istruzione ALTER DATASHARE. In tal caso, gli utenti Account

AWS che hanno accesso al datashare perdono l'accesso se i loro cluster sono accessibili pubblicamente. L'esecuzione di questo tipo di alterazione può violare i termini del prodotto dei dati in AWS Data Exchange. Per un'eccezione a questa raccomandazione, vedere di seguito.

L'esempio seguente mostra un errore quando viene creato un AWS Data Exchange datashare con l'impostazione disattivata.

```
ALTER DATASHARE salesshare SET PUBLICACCESSIBLE FALSE;
ERROR:  Alter of ADX-managed datashare salesshare requires session variable
 datashare_break_glass_session_var to be set to value 'c670ba4db22f4b'
```

Per consentire la modifica di un AWS Data Exchange datashare per disattivare l'impostazione accessibile al pubblico, impostate la variabile seguente ed eseguite nuovamente l'istruzione ALTER DATASHARE.

```
SET datashare_break_glass_session_var to 'c670ba4db22f4b';
```

```
ALTER DATASHARE salesshare SET PUBLICACCESSIBLE FALSE;
```

In questo caso, Amazon Redshift genera un valore casuale una tantum per impostare la variabile di sessione per consentire ALTER DATASHARE SET PUBLICACCESSIBLE FALSE per un'unità unità di condivisione dati AWS Data Exchange .

Esempi

L'esempio seguente aggiunge lo schema al datashare. `public` `salesshare`

```
ALTER DATASHARE salesshare ADD SCHEMA public;
```

Nell'esempio seguente vengono aggiunte le tabelle `public.tickit_sales_redshift` all'unità di condivisione dati `salesshare`.

```
ALTER DATASHARE salesshare ADD TABLE public.tickit_sales_redshift;
```

Nell'esempio seguente vengono aggiunte tutte le tabelle all'unità di condivisione dati `salesshare`

```
ALTER DATASHARE salesshare ADD ALL TABLES IN SCHEMA PUBLIC;
```

Nell'esempio seguente vengono rimosse le tabelle `public.tickit_sales_redshift` dall'unità di condivisione dati `salesshare`.

```
ALTER DATASHARE salesshare REMOVE TABLE public.tickit_sales_redshift;
```

ALTER DEFAULT PRIVILEGES

Definisce il set predefinito di autorizzazioni di accesso da applicare agli oggetti che verranno creati in futuro dall'utente specificato. Per impostazione predefinita, gli utenti possono modificare solo le proprie autorizzazioni di accesso di default. Solo un superutente può specificare le autorizzazioni predefinite per altri utenti.

È possibile applicare i privilegi predefiniti a ruoli, utenti o a gruppi di utenti. È possibile impostare le autorizzazioni predefinite a livello globale per tutti gli oggetti creati nel database corrente o per gli oggetti creati solo negli schemi specificati.

Le autorizzazioni predefinite si applicano solo ai nuovi oggetti. L'esecuzione di ALTER DEFAULT PRIVILEGES non modifica le autorizzazioni sugli oggetti esistenti. Per concedere autorizzazioni su tutti gli oggetti attuali e futuri creati da qualsiasi utente all'interno di un database o di uno schema, vedere Autorizzazioni con ambito.

Per visualizzare le informazioni sui privilegi predefiniti per gli utenti del database, eseguire una query sulla tabella del catalogo di sistema PG_DEFAULT_ACL.

Per ulteriori informazioni sui privilegi, consultare GRANT.

Privilegi richiesti

Di seguito sono elencati i privilegi richiesti per ALTER DEFAULT PRIVILEGES:

- Superuser
- Utenti con il privilegio ALTER DEFAULT PRIVILEGES
- Gli utenti che modificano i propri privilegi di accesso di default
- Utenti che definiscono i privilegi per gli schemi per i quali dispongono di privilegi di accesso

Sintassi

```
ALTER DEFAULT PRIVILEGES
    [ FOR USER target_user [, ...] ]
```

```
    [ IN SCHEMA schema_name [, ...] ]
    grant_or_revoke_clause

where grant_or_revoke_clause is one of:

GRANT { { SELECT | INSERT | UPDATE | DELETE | DROP | REFERENCES | TRUNCATE } [,...] |
 ALL [ PRIVILEGES ] }
 ON TABLES
 TO { user_name [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
 ON FUNCTIONS
 TO { user_name [ WITH GRANT OPTION ] |  ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
 ON PROCEDURES
 TO { user_name [ WITH GRANT OPTION ] |  ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]

REVOKE [ GRANT OPTION FOR ] { { SELECT | INSERT | UPDATE | DELETE | REFERENCES |
 TRUNCATE } [,...] | ALL [ PRIVILEGES ] }
 ON TABLES
 FROM user_name [, ...] [ RESTRICT ]

REVOKE  { { SELECT | INSERT | UPDATE | DELETE | REFERENCES | TRUNCATE } [,...] | ALL
 [ PRIVILEGES ] }
 ON TABLES
 FROM { ROLE role_name | GROUP group_name | PUBLIC } [, ...] [ RESTRICT ]

REVOKE [ GRANT OPTION FOR ] { EXECUTE | ALL [ PRIVILEGES ] }
 ON FUNCTIONS
 FROM user_name [, ...] [ RESTRICT ]

REVOKE { EXECUTE | ALL [ PRIVILEGES ] }
 ON FUNCTIONS
 FROM { ROLE role_name | GROUP group_name | PUBLIC } [, ...] [ RESTRICT ]

REVOKE [ GRANT OPTION FOR ] { EXECUTE | ALL [ PRIVILEGES ] }
 ON PROCEDURES
 FROM user_name [, ...] [ RESTRICT ]

REVOKE { EXECUTE | ALL [ PRIVILEGES ] }
```

```
ON PROCEDURES
FROM { ROLE role_name | GROUP group_name | PUBLIC } [, ...] [ RESTRICT ]
```

Parametri

FOR USER target_user

Facoltativo. Il nome dell'utente per il quale sono definiti i privilegi predefiniti. Solo l'utente con privilegi avanzati può specificare i privilegi predefiniti per gli altri utenti. Il valore di default è l'utente corrente.

IN SCHEMA schema_name

Facoltativo. Se viene visualizzata una clausola IN SCHEMA, i privilegi predefiniti specificati vengono applicati ai nuovi oggetti creati nel schema_name indicato. In questo caso, l'utente o il gruppo di utenti che costituisce il target di ALTER DEFAULT PRIVILEGES deve disporre del privilegio CREATE per lo schema specificato. I privilegi predefiniti specifici di uno schema vengono aggiunti ai privilegi predefiniti globali esistenti. Per impostazione predefinita, i privilegi predefiniti vengono applicati a livello globale all'intero database.

GRANT

L'insieme di privilegi da concedere agli utenti o ai gruppi specificati per tutte le nuove tabelle e viste, funzioni o stored procedure create dall'utente specificato. Con la clausola GRANT puoi impostare gli stessi privilegi e opzioni disponibili per il comando GRANT.

WITH GRANT OPTION

Clausola che indica che l'utente che riceve i privilegi può a sua volta concedere gli stessi privilegi agli altri. Non puoi concedere WITH GRANT OPTION a un gruppo o a PUBLIC.

TO user_name | ROLE role_name | GROUP group_name

Il nome dell'utente, del ruolo o del gruppo di utenti a cui vengono applicati i privilegi predefiniti specificati.

REVOKE

L'insieme di privilegi da revocare agli utenti o ai gruppi specificati per tutte le nuove tabelle, funzioni o procedure archiviate create dall'utente indicato. Con la clausola REVOKE puoi impostare gli stessi privilegi e opzioni disponibili per il comando REVOKE.

GRANT OPTION FOR

Clausola che revoca solo l'opzione per concedere un privilegio specificato ad altri utenti e non revoca il privilegio stesso. Non puoi revocare GRANT OPTION a un gruppo o a PUBLIC.

FROM user_name | ROLE role_name | GROUP group_name

Il nome dell'utente, del ruolo o del gruppo di utenti a cui vengono revocati per impostazione predefinita i privilegi specificati.

RESTRICT

L'opzione RESTRICT revoca solo i privilegi che l'utente ha concesso direttamente. Questa è l'impostazione predefinita.

Esempi

Si supponga di voler consentire a qualsiasi utente del gruppo di utenti di `report_readers` visualizzare tutte le tabelle e le viste create dall'utente. `report_admin` In questo caso, emettere il seguente comando come utente con privilegi avanzati.

```
alter default privileges for user report_admin grant select on tables to group
 report_readers;
```

Nell'esempio seguente, il primo comando concede i privilegi SELECT su tutte le nuove tabelle e viste create.

```
alter default privileges grant select on tables to public;
```

L'esempio seguente concede il privilegio INSERT al gruppo di utenti `sales_admin` per tutte le nuove tabelle e viste create nello schema `sales`.

```
alter default privileges in schema sales grant insert on tables to group sales_admin;
```

L'esempio seguente inverte il comando ALTER DEFAULT PRIVILEGES dell'esempio precedente.

```
alter default privileges in schema sales revoke insert on tables from group
 sales_admin;
```

Per impostazione predefinita, il gruppo di utenti PUBLIC dispone dell'autorizzazione execute per tutte le nuove funzioni definite dall'utente. Per revocare le autorizzazioni execute `public` per le nuove

funzioni e quindi concedere l'autorizzazione execute solo al gruppo di utenti `dev_test`, eseguire i seguenti comandi.

```
alter default privileges revoke execute on functions from public;
alter default privileges grant execute on functions to group dev_test;
```

ALTER EXTERNAL VIEW (anteprima)

Questa è una documentazione di pre-rilascio per le viste del Catalogo dati per Amazon Redshift, che è nella versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche . Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms.

È possibile creare un cluster Amazon Redshift di anteprima per testare nuove funzionalità di Amazon Redshift. Non è possibile utilizzare queste funzionalità in produzione o spostare il cluster di anteprima in un cluster di produzione o in un cluster su un'altra traccia. Per i termini e le condizioni dell'anteprima, consulta la sezione relativa a beta e anteprime nei termini del servizio AWS.

Come creare un cluster di anteprima

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegli Provisioned clusters dashboard (Pannello di controllo dei cluster con provisioning) e seleziona Clusters (Cluster). Regione AWS Sono elencati i cluster del tuo account nella versione corrente. Nelle colonne dell'elenco è visualizzato un sottoinsieme delle proprietà di ciascun cluster.
3. Nella pagina dell'elenco dei cluster viene visualizzato un banner che introduce l'anteprima. Scegli il pulsante Create preview cluster (Crea cluster di anteprima) per aprire la pagina di creazione del cluster.
4. Inserisci le proprietà del cluster. Scegli la traccia di anteprima che contiene le funzionalità che desideri testare. Consigliamo di assegnare al cluster un nome che indichi che si trova in una traccia di anteprima. Scegli le opzioni per il cluster, tra cui quelle contrassegnate come - anteprima, per le funzionalità che desideri testare. Per informazioni generali sulla creazione di cluster, consulta Creazione di un cluster nella Guida alla gestione di Amazon Redshift.
5. Per creare un cluster di anteprima, scegli Crea cluster.

> **Note**
>
> La traccia `preview_2023` è la traccia di anteprima più recente disponibile. Questa traccia supporta la creazione di cluster solo con tipi di nodo RA3. Il tipo di nodo DC2 e qualsiasi tipo di nodo precedente non sono supportati.

6. Quando il cluster di anteprima è disponibile, utilizza il client SQL per caricare dati ed eseguire query su di essi.

La funzionalità di anteprima delle viste del catalogo dati è disponibile solo nelle seguenti regioni.

- Stati Uniti orientali (Ohio) (us-east-2)
- Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Stati Uniti occidentali (California settentrionale) (us-west-1)
- Asia Pacifico (Tokyo) (ap-northeast-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Stoccolma) (eu-north-1)

Puoi anche creare un gruppo di lavoro di anteprima per testare le viste del catalogo dati. Non è possibile utilizzare queste funzionalità in produzione o trasferire il gruppo di lavoro in un altro gruppo di lavoro. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms. Per istruzioni su come creare un gruppo di lavoro di anteprima, consulta Creazione di un gruppo di lavoro di anteprima.

Utilizza il comando ALTER EXTERNAL VIEW per aggiornare la vista esterna. A seconda dei parametri utilizzati, possono essere interessati anche altri motori SQL, come Amazon Athena e Amazon EMR Spark, che possono fare riferimento a questa vista. Per ulteriori informazioni sulle viste del catalogo dati, consulta Creating Data Catalog views (preview).

Sintassi

```
ALTER EXTERNAL VIEW schema_name.view_name
{catalog_name.schema_name.view_name | awsdatacatalog.dbname.view_name |
 external_schema_name.view_name}
[FORCE] { AS (query_definition) | REMOVE DEFINITION }
```

Parametri

schema_name.view_name

Lo schema allegato al AWS Glue database, seguito dal nome della vista.

catalog_name.schema_name.view_name | awsdatacatalog.dbname.view_name | external_schema_name.view_name

La notazione dello schema da usare per la modifica della vista. È possibile specificare di utilizzare il AWS Glue Data Catalog, un database Glue creato dall'utente o uno schema esterno creato dall'utente. Per ulteriori informazioni, consulta CREATE DATABASE e CREATE EXTERNAL SCHEMA.

FORCE

Indica se AWS Lake Formation aggiornare la definizione della vista anche se gli oggetti a cui si fa riferimento nella tabella non sono coerenti con altri motori SQL. Se Lake Formation esegue l'aggiornamento, la vista viene considerata obsoleta per gli altri motori SQL fino a quando non vengono aggiornati.

AS query_definition

La definizione della query SQL che Amazon Redshift esegue per alterare la vista.

REMOVE DEFINITION

Indica se rilasciare e ricreare le viste. Le viste devono essere rilasciate e ricreate per contrassegnarle come `PROTECTED`.

Esempi

L'esempio seguente modifica una vista del catalogo dati denominata sample_schema.glue_data_catalog_view.

```
ALTER EXTERNAL VIEW sample_schema.glue_data_catalog_view
FORCE
REMOVE DEFINITION
```

ALTER FUNCTION

Rinomina una funzione o cambia il proprietario. Sono obbligatori sia il nome della funzione che i tipi di dati. Solo il proprietario o un superutente può rinominare una funzione. Solo un superutente può cambiare il proprietario di una funzione.

Sintassi

```
ALTER FUNCTION function_name ( { [ py_arg_name py_arg_data_type | sql_arg_data_type }
 [ , ... ] ] )
    RENAME TO new_name
```

```
ALTER FUNCTION function_name ( { [ py_arg_name py_arg_data_type | sql_arg_data_type }
 [ , ... ] ] )
    OWNER TO { new_owner | CURRENT_USER | SESSION_USER }
```

Parametri

function_name

Il nome della funzione da modificare. Specificate il nome della funzione nel percorso di ricerca corrente oppure utilizzate il formato `schema_name.function_name` per utilizzare uno schema specifico.

py_arg_name py_arg_data_type | sql_arg_data_type

Facoltativo. Un elenco di nomi di argomenti di input e tipi di dati per la funzione definita dall'utente Python o un elenco di tipi di dati degli argomenti di input per la funzione SQL definita dall'utente.

new_name

Un nuovo nome per la funzione definita dall'utente.

new_owner | CURRENT_USER | SESSION_USER

Un nuovo proprietario per la funzione definita dall'utente.

Esempi

L'esempio seguente modifica il nome di una funzione da `first_quarter_revenue` a `quarterly_revenue`.

```
ALTER FUNCTION first_quarter_revenue(bigint, numeric, int)
         RENAME TO quarterly_revenue;
```

L'esempio seguente modifica il proprietario della `quarterly_revenue` funzione in`etl_user`.

```
ALTER FUNCTION quarterly_revenue(bigint, numeric) OWNER TO etl_user;
```

ALTER GROUP

Cambia un gruppo di utenti. Usa questo comando per aggiungere utenti al gruppo, rimuovere utenti dal gruppo o rinominare il gruppo.

Sintassi

```
ALTER GROUP group_name
{
ADD USER username [, ... ] |
DROP USER username [, ... ] |
RENAME TO new_name
}
```

Parametri

group_name

Nome del gruppo di utenti da modificare.

ADD

Aggiunge un utente a un gruppo di utenti.

DROP

Rimuove un utente da un gruppo di utenti.

username

Nome dell'utente da aggiungere al gruppo o rimuovere dal gruppo.

RENAME TO

Rinomina il gruppo utente. I nomi di gruppo che iniziano con due caratteri di sottolineatura sono riservati all'uso interno di Amazon Redshift. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

new_name

Nuovo nome del gruppo di utenti.

Esempi

L'esempio seguente aggiunge un utente denominato DWUSER al gruppo ADMIN_GROUP.

```
ALTER GROUP admin_group
ADD USER dwuser;
```

L'esempio seguente rinomina il gruppo ADMIN_GROUP in ADMINISTRATORS.

```
ALTER GROUP admin_group
RENAME TO administrators;
```

L'esempio seguente aggiunge due utenti al gruppo ADMIN_GROUP.

```
ALTER GROUP admin_group
ADD USER u1, u2;
```

L'esempio seguente elimina due utenti dal gruppo ADMIN_GROUP.

```
ALTER GROUP admin_group
DROP USER u1, u2;
```

ALTER IDENTITY PROVIDER

Modifica un provider di identità per assegnare nuovi parametri e valori. Quando si esegue questo comando, tutti i valori dei parametri impostati in precedenza vengono eliminati prima dell'assegnazione dei nuovi valori. Solo un utente con privilegi avanzati può modificare un provider di identità.

Sintassi

```
ALTER IDENTITY PROVIDER identity_provider_name
[PARAMETERS parameter_string]
[NAMESPACE namespace]
[IAM_ROLE iam_role]
[DISABLE | ENABLE]
```

Parametri

identity_provider_name

Il nome del nuovo provider di identità. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

parameter_string

Stringa contenente un oggetto JSON formattato correttamente che contiene i parametri e i valori richiesti per il provider di identità specifico.

spazio dei nomi

Lo spazio dei nomi dell'organizzazione.

iam_role

Il ruolo IAM che fornisce le autorizzazioni per la connessione a IAM Identity Center. Questo parametro è applicabile solo quando il tipo di provider di identità è. AWSIDC

DISABILITA o ABILITA

Attiva o disattiva un provider di identità. L'impostazione predefinita è ENABLE

Esempi

Nell'esempio seguente viene modificato un provider di identità denominato oauth_standard. Si applica in particolare a quando Microsoft Azure AD è il provider di identità.

```
ALTER IDENTITY PROVIDER oauth_standard
PARAMETERS '{"issuer":"https://sts.windows.net/2sdfdsf-d475-420d-b5ac-667adad7c702/",
"client_id":"87f4aa26-78b7-410e-bf29-57b39929ef9a",
"client_secret":"BUAH~ewrqewrqwerUUY^%tHe1oNZShoiU7",
"audience":["https://analysis.windows.net/powerbi/connector/AmazonRedshift"]
}'
```

L'esempio seguente mostra come impostare lo spazio dei nomi del provider di identità. Ciò può essere applicato a Microsoft Azure AD, se segue un'istruzione come nell'esempio precedente, o a un altro provider di identità. Può essere applicato anche al caso in cui connetti un cluster con provisioning Amazon Redshift esistente o un gruppo di lavoro Serverless Amazon Redshift a IAM Identity Center, se disponi di una connessione configurata tramite un'applicazione gestita.

```
ALTER IDENTITY PROVIDER "my-redshift-idc-application"
NAMESPACE 'MYCO';
```

L'esempio seguente imposta il ruolo IAM e funziona nello use case per configurare l'integrazione di Redshift con IAM Identity Center.

```
ALTER IDENTITY PROVIDER "my-redshift-idc-application"
IAM_ROLE 'arn:aws:iam::123456789012:role/myadministratorrole';
```

Per ulteriori informazioni sulla configurazione di una connessione a IAM Identity Center da Redshift, consulta Connect Redshift with IAM Identity Center per offrire agli utenti un'esperienza di single sign-on.

Disabilitazione di un provider di identità

L'istruzione di esempio seguente mostra come disabilitare un provider di identità. Quando è disabilitato, gli utenti federati del provider di identità non possono accedere al cluster finché non viene nuovamente abilitato.

```
ALTER IDENTITY PROVIDER "redshift-idc-app" DISABLE;
```

ALTER MASKING POLICY

Modifica una politica di mascheramento dinamico dei dati esistente. Per ulteriori informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.

Una politica di mascheramento può essere modificata da utenti con privilegi avanzati e da utenti o ruoli che dispongono del ruolo sys:secadmin.

Sintassi

```
ALTER MASKING POLICY policy_name
   USING (masking_expression);
```

Parametri

nome_policy

Nome della policy di mascheramento. Deve essere il nome di una politica di mascheramento già esistente nel database.

masking_expression

Espressione SQL utilizzata per trasformare le colonne di destinazione. Può essere scritta utilizzando funzioni di manipolazione dei dati, come le funzioni di manipolazione delle stringhe, o in combinazione con funzioni definite dall'utente scritte in SQL, Python o con AWS Lambda.

L'espressione deve corrispondere alle colonne di input e ai tipi di dati dell'espressione originale. Ad esempio, se le colonne di input della politica di mascheramento originale fossero `sample_1 FLOAT` e `sample_2 VARCHAR(10)`, non sarebbe possibile modificare la politica di mascheramento per prendere una terza colonna o fare in modo che la politica assuma un valore FLOAT e un valore BOOLEAN. Se si utilizza una costante come espressione di mascheramento, è necessario convertirla in modo esplicito su un tipo che corrisponda al tipo di input.

È necessario disporre dell'autorizzazione USAGE per tutte le funzioni definite dall'utente utilizzate nell'espressione di mascheramento.

ALTER MATERIALIZED VIEW

Consente l'aggiornamento automatico di una vista materializzata

Sintassi

```
ALTER MATERIALIZED VIEW mv_name
[ AUTO REFRESH { YES | NO } ]
[ ROW LEVEL SECURITY { ON | OFF } [ CONJUNCTION TYPE { AND | OR } ] [FOR DATASHARES] ];
```

Parametri

mv_name

Il nome della vista materializzata da modificare.

AUTO REFRESH { YES | NO }

Una clausola che attiva o disattiva l'aggiornamento automatico di una vista materializzata. Per informazioni sull'aggiornamento automatico delle viste materializzate, consultare Aggiornamento di una vista materializzata.

ROW LEVEL SECURITY { ON | OFF } [CONJUNCTION TYPE { AND | OR }] [FOR DATASHARES]

Una clausola che attiva o disattiva la sicurezza a livello di riga per una relazione.

Quando per una relazione è attivata la protezione a livello di riga, è possibile leggere solo le righe al livello di riga a cui la policy di sicurezza consente l'accesso. Quando non ci sono policy che consentono l'accesso alla relazione, non è possibile visualizzare alcuna riga dalla relazione. Solo gli utenti con privilegi avanzati e gli utenti o i ruoli che hanno il ruolo `sys:secadmin` possono impostare la clausola ROW LEVEL SECURITY. Per ulteriori informazioni, consulta Sicurezza a livello di riga.

- [CONJUNCTION TYPE { AND | OR }]

  Una clausola che consente di scegliere il tipo di congiunzione della policy di sicurezza a livello di riga per una relazione. Quando a una relazione sono associate più policy di sicurezza a livello di riga, è possibile combinare le policy con la clausola AND oppure OR. Per impostazione predefinita, Amazon Redshift combina le policy RLS con la clausola AND. Gli utenti con privilegi avanzati, gli utenti o i ruoli che hanno il ruolo `sys:secadmin` possono utilizzare questa clausola per definire il tipo di combinazione della policy di sicurezza a livello di riga per una relazione. Per ulteriori informazioni, consulta Combinazione di più policy per utente.

- FOR DATASHARES

  Una clausola che determina se è possibile accedere a una relazione protetta da RLS in un'unità di condivisione dati. Per impostazione predefinita, non è possibile accedere a una relazione protetta da RLS in un'unità di condivisione dati. Il comando ALTER MATERIALIZED VIEW ROW LEVEL SECURITY eseguito con questa clausola influisce solo sulla proprietà di accessibilità dell'unità di condivisione dati della relazione. La proprietà ROW LEVEL SECURITY non viene modificata.

  Se rendi accessibile una relazione protetta da RLS nelle unità di condivisione dati, la relazione non dispone di una sicurezza a livello di riga nel database con unità di condivisione dati sul lato consumer. La relazione mantiene la proprietà RLS sul lato producer.

Esempi

Nell'esempio seguente la vista materializzata `tickets_mv` viene abilitata perché possa essere aggiornata automaticamente.

```
ALTER MATERIALIZED VIEW tickets_mv AUTO REFRESH YES
```

Esempi in DISTSTYLE e SORTKEY

Gli esempi in questo argomento mostrano come eseguire modifiche a DISTSTYLE e SORTKEY utilizzando ALTER MATERIALIZED VIEW.

Le seguenti query di esempio mostrano come modificare una colonna DISTSTYLE KEY DISTKEY utilizzando una tabella base di esempio:

```
CREATE TABLE base_inventory(
inv_date_sk int4 not null,
inv_item_sk int4 not null,
inv_warehouse_sk int4 not null,
inv_quantity_on_hand int4
);

INSERT INTO base_inventory VALUES(1,1,1,1);

CREATE MATERIALIZED VIEW inventory DISTSTYLE EVEN
as SELECT * FROM base_inventory;
SELECT "table", DISTSTYLE FROM svv_table_info WHERE "table" = 'inventory';

ALTER MATERIALIZED VIEW inventory ALTER DISTSTYLE KEY DISTKEY inv_warehouse_sk;
SELECT "table", DISTSTYLE FROM svv_table_info where "table" = 'inventory';

ALTER MATERIALIZED VIEW inventory ALTER DISTKEY inv_item_sk;
SELECT "table", diststyle from svv_table_info where "table" = 'inventory';
```

Modificare una vista materializzata in DISTSTYLE ALL:

```
CREATE TABLE base_inventory(
inv_date_sk int4 not null,
inv_item_sk int4 not null,
inv_warehouse_sk int4 not null,
inv_quantity_on_hand int4
);

INSERT INTO base_inventory values(1,1,1,1);

CREATE MATERIALIZED VIEW inventory DISTSTYLE EVEN
as SELECT * FROM base_inventory;

SELECT "table", DISTSTYLE FROM svv_table_info WHERE "table" = 'inventory';
```

I comandi seguenti mostrano esempi di ALTER MATERIALIZED VIEW SORTKEY, utilizzando una tabella base di esempio:

```
CREATE MATERIALIZED VIEW base_inventory (c0 int, c1 int);

CREATE MATERIALIZED VIEW inventory
interleaved sortkey(c0, c1)
as SELECT * FROM base_inventory;

SELECT "table", sortkey1 FROM svv_table_info WHERE "table" = 'inventory';

ALTER MATERIALIZED VIEW t1 alter sortkey(c0, c1);
SELECT "table", diststyle, sortkey_num FROM svv_table_info WHERE "table" = 'inventory';

ALTER MATERIALIZED VIEW t1 alter sortkey none;
SELECT "table", diststyle, sortkey_num FROM svv_table_info WHERE "table" = 'inventory';

ALTER MATERIALIZED VIEW t1 alter sortkey(c0);
SELECT "table", diststyle, sortkey_num FROM svv_table_info WHERE "table" = 'inventory';
```

ALTER RLS POLICY

Modifica di una policy di sicurezza a livello di riga esistente su una tabella.

Una policy può essere modificata da un utente con privilegi avanzati e da utenti o ruoli che dispongono del ruolo `sys:secadmin`.

Sintassi

```
ALTER RLS POLICY policy_name
USING ( using_predicate_exp );
```

Parametri

nome_policy

Il nome della policy .

USING (using_predicate_exp)

Specifica un filtro applicato alla clausola WHERE della query. Amazon Redshift applica un predicato di policy prima dei predicati utente a livello di query. Ad esempio, **`current_user =`**

`'joe' and price > 10` limita Joe a visualizzare solo i record con un prezzo superiore a 10 USD.

L'espressione ha accesso alle variabili dichiarate nella clausola WITH dell'istruzione CREATE RLS POLICY utilizzata per creare la policy con il nome "nome_policy".

Esempi

L'esempio seguente modifica una policy RLS.

```
-- First create an RLS policy that limits access to rows where catgroup is 'concerts'.
CREATE RLS POLICY policy_concerts
WITH (catgroup VARCHAR(10))
USING (catgroup = 'concerts');

-- Then, alter the RLS policy to only show rows where catgroup is 'piano concerts'.
ALTER RLS POLICY policy_concerts
USING (catgroup = 'piano concerts');
```

ALTER ROLE

Rinomina un ruolo o cambia il proprietario. Per un elenco di ruoli Amazon Redshift definiti dal sistema, consulta the section called “Ruoli definiti dal sistema di Amazon Redshift”.

Autorizzazioni richieste

Di seguito sono riportate le autorizzazioni richieste per ALTER ROLE:

- Superuser
- Utenti con le autorizzazioni ALTER ROLE

Sintassi

```
ALTER ROLE role [ WITH ]
  { { RENAME TO role } | { OWNER TO user_name } }[, ...]
  [ EXTERNALID TO external_id ]
```

Parametri

ruolo

Il nome del ruolo da modificare.

RENAME TO

Un nuovo nome per il ruolo.

OWNER TO user_name

Un nuovo proprietario per il ruolo.

EXTERNALID TO external_id

Un nuovo ID esterno per il ruolo, associato a un provider di identità. Per ulteriori informazioni, consulta Native identity provider (IdP) federation for Amazon Redshift (Federazione di provider di identità nativi (IdP) per Amazon Redshift).

Esempi

L'esempio seguente cambia il nome di un ruolo da `sample_role1` a `sample_role2`.

```
ALTER ROLE sample_role1 WITH RENAME TO sample_role2;
```

L'esempio seguente cambia il proprietario del ruolo.

```
ALTER ROLE sample_role1 WITH OWNER TO user1
```

La sintassi di ALTER ROLE è simile ad ALTER PROCEDURE di seguito.

```
ALTER PROCEDURE first_quarter_revenue(bigint, numeric) RENAME TO quarterly_revenue;
```

L'esempio seguente cambia il proprietario di una procedura in `etl_user`.

```
ALTER PROCEDURE quarterly_revenue(bigint, numeric) OWNER TO etl_user;
```

Nell'esempio seguente viene aggiornato un ruolo `sample_role1` con un nuovo ID esterno associato a un provider di identità.

```
ALTER ROLE sample_role1 EXTERNALID TO "XYZ456";
```

ALTER PROCEDURE

Rinomina una procedura o cambia il proprietario. Sono richiesti sia il nome della procedura che i tipi di dati o la firma. Una procedura può essere rinominata solo dal proprietario o da un utente con privilegi avanzati. Solo un utente con privilegi avanzati può cambiare il proprietario di una procedura.

Sintassi

```
ALTER PROCEDURE sp_name [ ( [ [ argname ] [ argmode ] argtype [, ...] ] ) ]
    RENAME TO new_name
```

```
ALTER PROCEDURE sp_name [ ( [ [ argname ] [ argmode ] argtype [, ...] ] ) ]
    OWNER TO { new_owner | CURRENT_USER | SESSION_USER }
```

Parametri

sp_name

Il nome della procedura da cambiare. Specifica solo il nome della procedura nel percorso di ricerca corrente oppure utilizza il formato `schema_name.sp_procedure_name` per utilizzare uno schema specifico.

[argname] [argmode] argtype

Elenco di nomi di argomento, modalità di argomento e tipi di dati. Sono richiesti solo i tipi di dati di input, utilizzati per individuare la procedura archiviata. In alternativa, puoi fornire la firma completa utilizzata per creare la procedura che include i parametri di input e output con le relative modalità.

new_name

Nuovo nome per la procedura archiviata.

new_owner | CURRENT_USER | SESSION_USER

Nuovo proprietario della procedura archiviata.

Esempi

L'esempio seguente cambia il nome di una procedura da `first_quarter_revenue` a `quarterly_revenue`.

```
ALTER PROCEDURE first_quarter_revenue(volume INOUT bigint, at_price IN numeric,
```

```
 result OUT int) RENAME TO quarterly_revenue;
```

Questo esempio è uguale al seguente.

```
ALTER PROCEDURE first_quarter_revenue(bigint, numeric) RENAME TO quarterly_revenue;
```

L'esempio seguente cambia il proprietario di una procedura in `etl_user`.

```
ALTER PROCEDURE quarterly_revenue(bigint, numeric) OWNER TO etl_user;
```

ALTER SCHEMA

Cambia la definizione di uno schema esistente. Usa questo comando per rinominare uno schema o modificare il proprietario di uno schema. Ad esempio, rinomina uno schema esistente per conservare una copia di backup di tale schema quando pianifichi di creare una nuova versione dello schema. Per ulteriori informazioni sugli schemi, consultare CREATE SCHEMA.

Per visualizzare le quote dello schema configurate, consultare SVV_SCHEMA_QUOTA_STATE.

Per visualizzare i record in cui le quote dello schema sono state superate, consultare STL_SCHEMA_QUOTA_VIOLATIONS.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per ALTER SCHEMA:

- Superuser
- Utente con il privilegio ALTER SCHEMA
- Proprietario dello schema

Quando si modifica il nome di uno schema, tenere presente che gli oggetti che utilizzano il vecchio nome, come le procedure archiviate o le viste materializzate, devono essere aggiornati per utilizzare il nuovo nome.

Sintassi

```
ALTER SCHEMA schema_name
```

```
{
RENAME TO new_name |
OWNER TO new_owner |
QUOTA { quota [MB | GB | TB] | UNLIMITED }
}
```

Parametri

schema_name

Il nome dello schema del database da modificare.

RENAME TO

Clausola che rinomina lo schema.

new_name

Il nuovo nome dello schema. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

OWNER TO

Clausola che modifica il proprietario dello schema.

new_owner

Il nuovo proprietario dello schema.

QUOTA

La quantità massima di spazio su disco che lo schema specificato può utilizzare. Questo spazio è la dimensione collettiva di tutte le tabelle nello schema specificato. Amazon Redshift converte il valore selezionato in megabyte. Il gigabyte è l'unità di misura predefinita quando non si specifica un valore.

Per ulteriori informazioni sulla configurazione delle quote dello schema, consultare CREATE SCHEMA.

Esempi

L'esempio seguente rinomina lo schema SALES in US_SALES.

```
alter schema sales
```

```
rename to us_sales;
```

L'esempio seguente concede la proprietà dello schema US_SALES all'utente DWUSER.

```
alter schema us_sales
owner to dwuser;
```

Nell'esempio seguente la quota viene modificata a 300 GB e viene rimossa.

```
alter schema us_sales QUOTA 300 GB;
alter schema us_sales QUOTA UNLIMITED;
```

ALTER SYSTEM

Modifica un'opzione di configurazione a livello di sistema per il cluster Amazon Redshift o il gruppo di lavoro Redshift Serverless.

Privilegi richiesti

Uno dei seguenti tipi di utente può eseguire il comando ALTER SYSTEM:

- Superuser
- Utente amministratore

Sintassi

```
ALTER SYSTEM SET system-level-configuration = {true| t | on | false | f | off}
```

Parametri

system-level-configuration

Una configurazione a livello di sistema. Valori validi: `data_catalog_auto_mount` e `metadata_security`.

{vero| t | attivo | falso | f | disattivato}

Un valore per attivare o disattivare la configurazione a livello di sistema.
Un `true`, `t` oppure `on` indica di attivare la configurazione. Un `false`, `f` oppure `off` indica di disattivare la configurazione.

Note per l'utilizzo

Per un cluster con provisioning, le modifiche apportate a `data_catalog_auto_mount` diventano effettive al successivo riavvio del cluster. Per ulteriori informazioni, consulta Riavvio di un cluster nella Guida alla gestione di Amazon Redshift.

Per un gruppo di lavoro serverless, le modifiche apportate a `data_catalog_auto_mount` non diventano immediatamente effettive.

Esempi

L'esempio seguente attiva il montaggio automatico di AWS Glue Data Catalog.

```
ALTER SYSTEM SET data_catalog_auto_mount = true;
```

L'esempio seguente attiva la sicurezza dei metadati.

```
ALTER SYSTEM SET metadata_security = true;
```

Impostazione di uno spazio dei nomi di identità predefinito

Questo esempio è specifico per l'utilizzo di un provider di identità. Puoi integrare Redshift con IAM Identity Center e un provider di identità per centralizzare la gestione delle identità per Redshift e altri servizi. AWS

L'esempio seguente mostra come impostare lo spazio dei nomi di identità predefinito per il sistema. Questa operazione successivamente semplifica l'esecuzione delle istruzioni GRANT e CREATE, poiché non è necessario includere lo spazio dei nomi come prefisso per ogni identità.

```
ALTER SYSTEM SET default_identity_namespace = 'MYCO';
```

Dopo aver eseguito il comando, è possibile eseguire istruzioni come le seguenti:

```
GRANT SELECT ON TABLE mytable TO alice;

GRANT UPDATE ON TABLE mytable TO salesrole;

CREATE USER bob password 'md50c983d1a624280812631c5389e60d48c';
```

L'effetto dell'impostazione dello spazio dei nomi di identità predefinito è che ogni identità non lo richiede come prefisso. In questo esempio, `alice` viene sostituito da. `MYCO:alice` Ciò accade con

qualsiasi identità inclusa. Per ulteriori informazioni sull'utilizzo di un provider di identità con Redshift, consulta Connect Redshift with IAM Identity Center per offrire agli utenti un'esperienza di single sign-on.

Per ulteriori informazioni sulle impostazioni relative alla configurazione di Redshift con IAM Identity Center, SET consulta e. ALTER IDENTITY PROVIDER

ALTER TABLE

Questo comando modifica la definizione di una tabella del database Amazon Redshift o la tabella esterna di Amazon Redshift Spectrum. Questo comando aggiorna i valori e le proprietà impostati da CREATE TABLE o CREATE EXTERNAL TABLE.

Non è possibile eseguire ALTER TABLE su una tabella esterna all'interno di un blocco di transazione (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

ALTER TABLE blocca le operazioni di lettura e scrittura della tabella fino al completamento della transazione che contiene l'operazione ALTER TABLE, a meno che nella documentazione non sia specificamente indicato che è possibile eseguire query sui dati o eseguire altre operazioni sulla tabella durante la modifica.

Privilegi richiesti

L'utente che modifica una tabella deve disporre del privilegio appropriato affinché il comando venga eseguito correttamente. A seconda del comando ALTER TABLE, è richiesto uno dei seguenti privilegi.

- Superuser
- Utenti con il privilegio ALTER TABLE
- Il proprietario della tabella con il privilegio USAGE sullo schema

Sintassi

```
ALTER TABLE table_name
{
ADD table_constraint
| DROP CONSTRAINT constraint_name [ RESTRICT | CASCADE ]
| OWNER TO new_owner
| RENAME TO new_name
| RENAME COLUMN column_name TO new_name
```

```
| ALTER COLUMN column_name TYPE updated_varchar_data_type_size
| ALTER COLUMN column_name ENCODE new_encode_type
| ALTER COLUMN column_name ENCODE encode_type,
| ALTER COLUMN column_name ENCODE encode_type, .....;
| ALTER DISTKEY column_name
| ALTER DISTSTYLE ALL
| ALTER DISTSTYLE EVEN
| ALTER DISTSTYLE KEY DISTKEY column_name
| ALTER DISTSTYLE AUTO
| ALTER [COMPOUND] SORTKEY ( column_name [,...] )
| ALTER SORTKEY AUTO
| ALTER SORTKEY NONE
| ALTER ENCODE AUTO
| ADD [ COLUMN ] column_name column_type
  [ DEFAULT default_expr ]
  [ ENCODE encoding ]
  [ NOT NULL | NULL ]
  [ COLLATE { CASE_SENSITIVE | CASE_INSENSITIVE } ] |
| DROP [ COLUMN ] column_name [ RESTRICT | CASCADE ]
| ROW LEVEL SECURITY { ON | OFF } [ CONJUNCTION TYPE { AND | OR } ] [ FOR DATASHARES ]}

where table_constraint is:

[ CONSTRAINT constraint_name ]
{ UNIQUE ( column_name [, ... ] )
| PRIMARY KEY ( column_name [, ... ] )
| FOREIGN KEY (column_name [, ... ] )
   REFERENCES  reftable [ ( refcolumn ) ]}

The following options apply only to external tables:

SET LOCATION { 's3://bucket/folder/' | 's3://bucket/manifest_file' }
| SET FILE FORMAT format |
| SET TABLE PROPERTIES ('property_name'='property_value')
| PARTITION ( partition_column=partition_value [, ...] )
  SET LOCATION { 's3://bucket/folder' |'s3://bucket/manifest_file' }
| ADD [IF NOT EXISTS]
    PARTITION ( partition_column=partition_value [, ...] ) LOCATION
 { 's3://bucket/folder' |'s3://bucket/manifest_file' }
    [, ... ]
| DROP PARTITION ( partition_column=partition_value [, ...] )
```

Per ridurre il tempo necessario per eseguire il comando ALTER TABLE, è possibile combinare alcune clausole del comando ALTER TABLE.

Amazon Redshift supporta le seguenti combinazioni delle clausole ALTER TABLE:

```
ALTER TABLE tablename ALTER SORTKEY (column_list), ALTER DISTKEY column_Id;
ALTER TABLE tablename ALTER DISTKEY column_Id, ALTER SORTKEY (column_list);
ALTER TABLE tablename ALTER SORTKEY (column_list), ALTER DISTSTYLE ALL;
ALTER TABLE tablename ALTER DISTSTYLE ALL, ALTER SORTKEY (column_list);
```

Parametri

table_name

Nome della tabella da modificare. Specifica solo il nome della tabella o utilizzare il formato schema_name.table_name per utilizzare uno schema specifico. Le tabelle esterne devono essere qualificate da un nome di schema esterno. Inoltre puoi specificare un nome di vista se utilizzi l'istruzione ALTER TABLE per rinominare una vista o cambiare il proprietario. La lunghezza massima per il nome della tabella è 127 byte; i nomi più lunghi vengono troncati a 127 byte. Puoi utilizzare caratteri multibyte UTF-8 fino a un massimo di quattro byte. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

ADD table_constraint

Clausola che aggiunge il vincolo specificato alla tabella. Per le descrizioni dei valori validi di table_constraint, vedi CREATE TABLE.

> **Note**
>
> Non puoi aggiungere un vincolo della chiave primaria a una colonna nullable. Se la colonna è stata originariamente creata con il vincolo NOT NULL, puoi aggiungere il vincolo della chiave primaria.

DROP CONSTRAINT constraint_name

Clausola che rimuove il vincolo denominato dalla tabella. Per rimuovere un vincolo, specifica il nome del vincolo e non il tipo di vincolo. Per visualizzare i nomi dei vincoli di tabella, esegui la seguente query.

```
select constraint_name, constraint_type
```

```
from information_schema.table_constraints;
```

RESTRICT

Clausola che rimuove solo il vincolo specificato. RESTRICT è un'opzione per DROP CONSTRAINT. RESTRICT non può essere usata con CASCADE.

CASCADE

Clausola che rimuove il vincolo specificato e tutto quello che dipende da quel vincolo. CASCADE è un'opzione per DROP CONSTRAINT. CASCADE non può essere usata con RESTRICT.

OWNER TO new_owner

Clausola che modifica il proprietario della tabella (o vista) sul valore new_owner.

RENAME TO new_name

Clausola che rinomina una tabella (o una vista) sul valore specificato in new_name. La lunghezza massima per il nome della tabella è 127 byte; i nomi più lunghi vengono troncati a 127 byte.

Non è possibile rinominare una tabella permanente con un nome che inizia con "#". Un nome di tabella che inizia con "#" indica una tabella temporanea.

Non puoi rinominare una tabella esterna.

ALTER COLUMN column_name TYPE updated_varchar_data_type_size

Una clausola che modifica le dimensioni di una colonna definita come un tipo di dati VARCHAR. Questa clausola supporta solo la modifica della dimensione di un tipo di dati VARCHAR. Considera i seguenti limiti:

- Non puoi modificare una colonna con le codifiche di compressione BYTEDICT, RUNLENGTH, TEXT255 o TEXT32K.
- Non è possibile ridurre le dimensioni al di sotto delle dimensioni massime dei dati esistenti.
- Non puoi alterare le colonne con valori predefiniti.
- Non puoi alterare le colonne con UNIQUE, PRIMARY KEY o FOREIGN KEY.
- Non è possibile modificare le colonne all'interno di un blocco di transazione (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

ALTER COLUMN column_name ENCODE new_encode_type

Una clausola che modifica la codifica di compressione di una colonna. Se si specifica la codifica di compressione per una colonna, la tabella non è più impostata su ENCODE AUTO. Per

informazioni sulla codifica della compressione, consultare Utilizzo della compressione delle colonne.

Quando si modifica la codifica di compressione per una colonna, la tabella rimane disponibile per l'esecuzione di query.

Considera i seguenti limiti:

- Non è possibile modificare una colonna con la stessa codifica definita per la colonna.
- Non è possibile modificare la codifica per una colonna in una tabella con un sortkey interlacciato.

ALTER COLUMN column_name ENCODE encode_type, ALTER COLUMN column_name ENCODE encode_type, ...;

Una clausola che modifica la codifica di compressione di più colonne in un singolo comando. Per informazioni sulla codifica della compressione, consultare Utilizzo della compressione delle colonne.

Quando si modifica la codifica di compressione per una colonna, la tabella rimane disponibile per l'esecuzione di query.

Considera i seguenti limiti:

- Non è possibile modificare più volte una colonna con lo stesso tipo di codifica o con un tipo di codifica diverso in un singolo comando.
- Non è possibile modificare una colonna con la stessa codifica definita per la colonna.
- Non è possibile modificare la codifica per una colonna in una tabella con un sortkey interlacciato.

ALTER DISTSTYLE ALL

Una clausola che modifica lo stile di distribuzione esistente di una tabella in ALL. Considera i seguenti aspetti:

- ALTER DISTSTYLE, ALTER SORTKEY e VACUUM non possono essere eseguiti contemporaneamente sulla stessa tabella.
  - Se VACUUM è attualmente in esecuzione, l'esecuzione di ALTER DISTSTYLE ALL restituisce un errore.
  - Se ALTER DISTSTYLE è in esecuzione, il vacuum di background non viene avviato su una tabella.

- Il comando ALTER DISTSTYLE ALL non è supportato per le tabelle con chiavi di ordinamento interlacciato e tabelle temporanee.
- Se lo stile di distribuzione è stato precedentemente definito come AUTO, la tabella non è più un candidato per l'ottimizzazione automatica della tabella.

Per ulteriori informazioni su DISTSTYLE ALL, consultare CREATE TABLE.

ALTER DISTSTYLE EVEN

Una clausola che modifica lo stile di distribuzione esistente di una tabella in EVEN. Considera i seguenti aspetti:

- ALTER DISTSYTLE, ALTER SORTKEY e VACUUM non possono essere eseguiti contemporaneamente sulla stessa tabella.
  - Se VACUUM è attualmente in esecuzione, l'esecuzione di ALTER DISTSTYLE EVEN restituisce un errore.
  - Se ALTER DISTSTYLE EVEN è in esecuzione, il vacuum di background non viene avviato su una tabella.
- Il comando ALTER DISTSTYLE EVEN non è supportato per le tabelle con chiavi di ordinamento interlacciato e tabelle temporanee.
- Se lo stile di distribuzione è stato precedentemente definito come AUTO, la tabella non è più un candidato per l'ottimizzazione automatica della tabella.

Per ulteriori informazioni su DISTSTYLE ALL EVEN, consultare CREATE TABLE.

ALTER DISTKEY column_name o ALTER DISTSTYLE KEY DISTKEY column_name

Una clausola che modifica la colonna utilizzata come chiave di distribuzione di una tabella. Considera i seguenti aspetti:

- VACUUM e ALTER DISTKEY non possono essere eseguiti contestualmente sulla stessa tabella.
  - Se VACUUM è già in esecuzione, ALTER DISTKEY restituisce un errore.
  - Se ALTER DISTKEY è in esecuzione, il vacuum di background non viene avviato su una tabella.
  - Se ALTER DISTKEY è in esecuzione, il vacuum in primo piano restituisce un errore.
- Su una tabella, è possibile eseguire solo un comando ALTER DISTKEY alla volta.
- Il comando ALTER DISTKEY non è supportato sulle tabelle con chiavi di ordinamento interlacciato.

- Se lo stile di distribuzione è stato precedentemente definito come AUTO, la tabella non è più un candidato per l'ottimizzazione automatica della tabella.

Quando si specifica DISTSTYLE KEY, i dati vengono distribuiti dai valori nella colonna DISTKEY. Per ulteriori informazioni su DISTSTYLE, consultare CREATE TABLE.

ALTER DISTSTYLE AUTO

Una clausola che modifica lo stile di distribuzione esistente di una tabella in AUTO.

Quando si modifica uno stile di distribuzione su AUTO, lo stile di distribuzione della tabella viene impostato come segue:

- Una piccola tabella con DISTSTYLE ALL è convertita in AUTO(ALL).
- Una piccola tabella con DISTSTYLE EVEN è convertita in AUTO(ALL).
- Una piccola tabella con DISTSTYLE KEY è convertita in AUTO(ALL).
- Una tabella grande con DISTSTYLE ALL è convertita in AUTO(ALL).
- Una tabella grande con DISTSTYLE EVEN è convertita in AUTO(ALL).
- Una tabella grande con DISTSTYLE KEY viene convertita in AUTO(KEY) e il DISTKEY viene conservato In questo caso, Amazon Redshift non apporta modifiche alla tabella.

Se Amazon Redshift determina che un nuovo stile o chiave di distribuzione migliorerà le prestazioni delle query, Amazon Redshift potrebbe modificare lo stile o la chiave di distribuzione della tabella in futuro. Ad esempio, Amazon Redshift potrebbe convertire una tabella con un DISTSTYLE di AUTO(KEY) in AUTO(EVEN) o viceversa. Per ulteriori informazioni sul comportamento in caso di modifica delle chiavi di distribuzione, tra cui la ridistribuzione dei dati e i blocchi, consulta i suggerimenti di Amazon Redshift Advisor.

Per ulteriori informazioni su DISTSTYLE AUTO, consultare CREATE TABLE.

Per visualizzare lo stile di distribuzione di una tabella, eseguire una query sulla vista del catalogo di sistema SVV_TABLE_INFO. Per ulteriori informazioni, consulta SVV_TABLE_INFO. Per visualizzare i suggerimenti di Amazon Redshift Advisor per le tabelle, eseguire una query sulla vista del catalogo di sistema SVV_ALTER_TABLE_RECOMMENDATIONS. Per ulteriori informazioni, consulta SVV_ALTER_TABLE_RECOMMENDATIONS. Per visualizzare le azioni intraprese da Amazon Redshift, eseguire una query sulla vista del catalogo di sistema SVL_AUTO_WORKER_ACTION. Per ulteriori informazioni, consulta SVL_AUTO_WORKER_ACTION.

ALTER [COMPOUND] SORTKEY (column_name [,...])

Una clausola che modifica o aggiunge la chiave di ordinamento utilizzata per una tabella.

Quando si modifica una chiave di ordinamento, la codifica di compressione delle colonne nella chiave di ordinamento nuova o originale può cambiare. Se nessuna codifica è definita esplicitamente per la tabella, allora Amazon Redshift assegna automaticamente le codifiche di compressione come segue:

- Le colonne definite come chiavi di ordinamento vengono assegnate alla compressione RAW.
- Le colonne definite come tipi di dati BOOLEAN, REAL o DOUBLE PRECISION vengono assegnate alla compressione RAW.
- Le colonne definite come SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIME, TIMETZ, TIMESTAMP o TIMESTAMPTZ sono assegnate alla compressione AZ64.
- Le colonne definite come CHAR o VARCHAR sono assegnate alla compressione LZO.

Considera i seguenti aspetti:

- È possibile definire un massimo di 400 colonne per una chiave di ordinamento per tabella.
- È possibile modificare una chiave di ordinamento interlacciata in una chiave di ordinamento composta o nessuna chiave di ordinamento. È possibile modificare una chiave di ordinamento interlacciata in una chiave di ordinamento composta o nessuna chiave di ordinamento.
- Se la chiave di ordinamento è stata precedentemente definita come AUTO, la tabella non è più un candidato per l'ottimizzazione automatica della tabella.
- Amazon Redshift consiglia di utilizzare la codifica RAW (senza compressione) per le colonne definite come chiavi di ordinamento. Quando si modifica una colonna per sceglierla come chiave di ordinamento, la compressione della colonna viene modificata in compressione RAW (senza compressione). Ciò può aumentare la quantità di archiviazione richiesta dalla tabella. L'aumento delle dimensioni della tabella dipende dalla definizione specifica della tabella e dal contenuto della tabella. Per ulteriori informazioni su questi componenti, consultare . Codifiche di compressione

Quando i dati vengono caricati in una tabella seguono l'ordine della chiave di ordinamento. Quando si modifica la chiave di ordinamento, Amazon Redshift riordina i dati. Per ulteriori informazioni su SORTKEY, consultare CREATE TABLE.

ALTER SORTKEY AUTO

Una clausola che modifica o aggiunge la chiave di ordinamento della tabella di destinazione ad AUTO.

Quando si modifica una chiave di ordinamento in AUTO, Amazon Redshift conserva la chiave di ordinamento esistente della tabella.

Se Amazon Redshift determina che una nuova chiave di ordinamento migliorerà le prestazioni delle query, Amazon Redshift potrebbe modificare la chiave di ordinamento della tabella in futuro.

Per ulteriori informazioni su SORTKEY AUTO, consultare CREATE TABLE.

Per visualizzare la chiave di ordinamento di una tabella, eseguire una query sulla vista del catalogo di sistema SVV_TABLE_INFO. Per ulteriori informazioni, consulta SVV_TABLE_INFO. Per visualizzare i suggerimenti di Amazon Redshift Advisor per le tabelle, eseguire una query sulla vista del catalogo di sistema SVV_ALTER_TABLE_RECOMMENDATIONS. Per ulteriori informazioni, consulta SVV_ALTER_TABLE_RECOMMENDATIONS. Per visualizzare le azioni intraprese da Amazon Redshift, eseguire una query sulla vista del catalogo di sistema SVL_AUTO_WORKER_ACTION. Per ulteriori informazioni, consulta SVL_AUTO_WORKER_ACTION.

ALTER SORTKEY NONE

Una clausola che rimuove la chiave di ordinamento della tabella di destinazione.

Se la chiave di ordinamento è stata precedentemente definita come AUTO, la tabella non è più un candidato per l'ottimizzazione automatica della tabella.

ALTER ENCODE AUTO

Una clausola che modifica il tipo di codifica delle colonne della tabella di destinazione in AUTO. Quando si modifica la codifica in AUTO, Amazon Redshift conserva il tipo di codifica esistente delle colonne nella tabella. Quindi, se Amazon Redshift determina che un nuovo tipo di codifica può migliorare le prestazioni delle query, Amazon Redshift può modificare il tipo di codifica delle colonne della tabella.

Se si modifica una o più colonne per specificare una codifica, Amazon Redshift non regola più automaticamente la codifica per tutte le colonne della tabella. Le colonne mantengono le impostazioni di codifica correnti.

Le seguenti azioni non influiscono sull'impostazione ENCODE AUTO per la tabella:

- Ridenominazione della tabella.
- Modifica dell'impostazione DISTSTYLE o SORTKEY per la tabella.
- Aggiunta o eliminazione di una colonna con un'impostazione ENCODE.

- Utilizzo dell'opzione COMPUPDATE del comando COPY. Per ulteriori informazioni, consulta Operazioni di caricamento dati.

Per visualizzare la codifica di una tabella, eseguire una query sulla vista del catalogo di sistema SVV_TABLE_INFO. Per ulteriori informazioni, consulta SVV_TABLE_INFO.

RENAME COLUMN column_name TO new_name

Clausola che rinomina una colonna sul valore specificato in new_name. La lunghezza massima per il nome della colonna è 127 byte; i nomi più lunghi vengono troncati a 127 byte. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

ADD [COLUMN] column_name

Clausola che aggiunge una colonna con il nome specificato alla tabella. Puoi aggiungere una sola colonna in ogni istruzione ALTER TABLE.

Non puoi aggiungere una colonna che sia la chiave di distribuzione (DISTKEY) o una chiave di ordinamento (SORTKEY) della tabella.

Non puoi utilizzare un comando ALTER TABLE ADD COLUMN per modificare i seguenti attributi di tabella e colonna:

- UNIQUE
- PRIMARY KEY
- REFERENCES (chiave esterna)
- IDENTITY o GENERATED BY DEFAULT AS IDENTITY

La lunghezza massima per il nome della colonna è 127 byte; i nomi più lunghi vengono troncati a 127 byte. Il numero massimo di colonne che puoi definire in una singola tabella è 1.600.

Le seguenti restrizioni si applicano quando si aggiunge una colonna a una tabella esterna:

- Non puoi aggiungere una colonna a una tabella esterna con i vincoli di colonna DEFAULT, ENCODE, NOT NULL o NULL.
- Non puoi aggiungere colonne a una tabella esterna definita utilizzando il formato file AVRO.
- Se le pseudocolonne sono abilitate, il numero massimo di colonne che è possibile definire in una singola tabella esterna è 1.598. Se le pseudocolonne non sono abilitate, il numero massimo di colonne che puoi definire in una singola tabella è 1.600.

Per ulteriori informazioni, consulta CREATE EXTERNAL TABLE.

column_type

Tipo dei dati della colonna da aggiungere. Per le colonne CHAR e VARCHAR, puoi utilizzare la parola chiave MAX invece di dichiarare una lunghezza massima. MAX imposta la lunghezza massima a 4.096 byte per CHAR o a 65.535 byte per VARCHAR. La dimensione massima di un oggetto di tipo GEOMETRY è di 1.048.447 byte.

Per informazioni sui tipi di dati supportati da Amazon Redshift, consultare Tipi di dati.

DEFAULT default_expr

Clausola che assegna un valore di dati predefinito per la colonna. Il tipo di dati di default_expr deve corrispondere al tipo di dati della colonna. Il valore DEFAULT deve essere un'espressione senza variabili. Le sottoquery, i riferimenti incrociati ad altre colonne della tabella corrente e le funzioni definite dall'utente non sono consentiti.

default_expr è utilizzato in qualsiasi operazione INSERT che non specifica un valore per la colonna. Se non viene specificato alcun valore predefinito, il valore predefinito per la colonna è null.

Se un'operazione COPY incontra un campo nullo su una colonna che ha un valore DEFAULT e un vincolo NOT NULL, il comando COPY inserisce il valore di default_expr.

DEFAULT non è supportato per le tabelle esterne.

ENCODE encoding

La codifica della compressione per una colonna. Per impostazione predefinita, Amazon Redshift gestisce automaticamente la codifica di compressione per tutte le colonne di una tabella se non si specifica la codifica di compressione per qualsiasi colonna della tabella o se si specifica l'opzione ENCODE AUTO per la tabella.

Se si specifica la codifica di compressione per qualsiasi colonna della tabella o se non si specifica l'opzione ENCODE AUTO per la tabella, Amazon Redshift assegna automaticamente la codifica di compressione alle colonne per le quali non si specifica la codifica di compressione come segue:

- Per impostazione predefinita, tutte le colonne nelle tabelle temporanee vengono assegnate alla compressione RAW.
- Le colonne definite come chiavi di ordinamento vengono assegnate alla compressione RAW.
- Alle colonne definite come tipi di dati BOOLEAN, REAL, DOUBLE PRECISION, GEOMETRY o GEOGRAPHY viene assegnata la compressione di tipo RAW.

- Le colonne definite come SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIME, TIMETZ, TIMESTAMP o TIMESTAMPTZ sono assegnate alla compressione AZ64.
- Le colonne definite come CHAR, VARCHAR o VARBYTE sono assegnate alla compressione LZO.

> **Note**
>
> Se non vuoi che una colonna venga compressa, specifica esplicitamente la codifica RAW.

Sono supportate le seguenti compression encodings (p. 68):

- AZ64
- BYTEDICT
- DELTA
- DELTA32K
- LZO
- MOSTLY8
- MOSTLY16
- MOSTLY32
- RAW (nessuna compressione)
- RUNLENGTH
- TEXT255
- TEXT32K
- ZSTD

ENCODE non è supportato per le tabelle esterne.

NOT NULL | NULL

NOT NULL specifica che la colonna non può contenere valori null. NULL, il valore predefinito, specifica che la colonna accetta valori nulli.

NOT NULL e NULL non sono supportati per tabelle esterne.

COLLATE { CASE_SENSITIVE | CASE_INSENSITIVE }

Una clausola che specifica se la stringa di ricerca o il confronto sulla colonna è CASE_SENSITIVE o CASE_INSENSITIVE. Il valore di default corrisponde alla stessa configurazione corrente della distinzione tra maiuscole e minuscole del database.

Per informazioni sul confronto di database, utilizzare il comando seguente:

```
SELECT db_collation();

db_collation
----------------
 case_sensitive
(1 row)
```

DROP [COLUMN] column_name

Il nome della colonna da eliminare dalla tabella.

Non puoi rimuovere l'ultima colonna di una tabella. Una tabella deve avere almeno una colonna.

Non puoi rimuovere una colonna che sia la chiave di distribuzione (DISTKEY) o una chiave di ordinamento (SORTKEY) della tabella. Il comportamento predefinito per DROP COLUMN è RESTRICT se la colonna contiene oggetti dipendenti, ad esempio una vista, una chiave primaria, una chiave esterna o una restrizione UNIQUE.

Le seguenti restrizioni si applicano quando si rimuove una colonna da una tabella esterna:

- Non puoi rimuovere una colonna da una tabella esterna se la colonna viene utilizzata come partizione.
- Non puoi rimuovere una colonna da una tabella esterna definita utilizzando il formato file AVRO.
- RESTRICT e CASCADE vengono ignorati per le tabelle esterne.
- Non è possibile eliminare le colonne della tabella delle policy a cui viene fatto riferimento all'interno della definizione della policy a meno che non si elimini o si scolleghi la policy. Ciò vale anche quando viene specificata l'opzione CASCADE. È possibile eliminare altre colonne nella tabella delle policy.

Per ulteriori informazioni, consulta CREATE EXTERNAL TABLE.

RESTRICT

Quando viene utilizzato con DROP COLUMN, RESTRICT indica che la colonna da rimuovere non è stata rimossa, in questi casi:

- Se una vista definita fa riferimento alla colonna che si vuole eliminare
- Se una chiave esterna fa riferimento alla colonna
- Se la colonna fa parte di una chiave multiparte

RESTRICT non può essere usata con CASCADE.

RESTRICT e CASCADE vengono ignorati per le tabelle esterne.

CASCADE

Se utilizzato con DROP COLUMN, rimuove la colonna specificata e tutto ciò che dipende da tale colonna. CASCADE non può essere usata con RESTRICT.

RESTRICT e CASCADE vengono ignorati per le tabelle esterne.

Le seguenti opzioni si applicano solo alle tabelle esterne.

SET LOCATION { 's3://bucket/folder/' | 's3://bucket/manifest_file' }

Il percorso della cartella Amazon S3 che contiene i file di dati o un file manifest che contiene un elenco di percorsi di oggetti Amazon S3. I bucket devono trovarsi nella stessa regione AWS del cluster Amazon Redshift. Per un elenco delle AWS regioni supportate, consulta. Considerazioni su Amazon Redshift Spectrum Per ulteriori informazioni sull'uso di un file manifest, vedi LOCATION nel riferimento CREATE EXTERNAL TABLE Parametri.

SET FILE FORMAT format

Il formato dei file di dati esterni.

I formati validi sono:

- AVRO
- PARQUET
- RCFILE
- SEQUENCEFILE

- TEXTFILE

SET TABLE PROPERTIES ('property_name'='property_value')

Clausola che imposta la definizione della tabella delle proprietà della tabella per una tabella esterna.

> **Note**
>
> Le proprietà della tabella rispettano la distinzione tra maiuscole e minuscole.

'numRows'='row_count'

Proprietà che imposta il valore numRows per la definizione della tabella. Per aggiornare in modo esplicito le statistiche di una tabella esterna, imposta la proprietà numRows in modo da indicare le dimensioni della tabella. Amazon Redshift non analizza le tabelle esterne per generare le statistiche delle tabelle che l'ottimizzatore di query utilizza per generare un piano di query. Se le statistiche della tabella non sono impostate per una tabella esterna, Amazon Redshift genera un piano di esecuzione della query. Il piano si basa sul presupposto che le tabelle esterne sono le tabelle più grandi e che quelle locali sono le più piccole.

'skip.header.line.count'='line_count'

Proprietà che imposta il numero di righe da saltare all'inizio di ogni file sorgente.

PARTITION (partition_column=partition_value [, ...] SET LOCATION { 's3://bucket/folder' | 's3://bucket/manifest_file' }

Clausola che imposta una nuova posizione per una o più colonne di partizione.

ADD [IF NOT EXISTS] PARTITION (partition_column=partition_value [, ...]) LOCATION { 's3://bucket/folder' | 's3://bucket/manifest_file' } [, ...]

Una clausola che aggiunge una o più partizioni. È possibile specificare più clausole PARTITION utilizzando una singola istruzione ALTER TABLE … ADD.

> **Note**
>
> Se si utilizza il AWS Glue catalogo, è possibile aggiungere fino a 100 partizioni utilizzando una singola istruzione ALTER TABLE.

La clausola IF NOT EXISTS indica che se la partizione specificata esiste già, il comando non deve apportare modifiche. Indica inoltre che il comando deve restituire un messaggio che la partizione esiste, anziché terminare con un errore. Questa clausola è utile durante lo scripting, quindi lo script non fallisce se ALTER TABLE tenta di aggiungere una partizione già esistente.

DROP PARTITION (partition_column=partition_value [, ...])

Clausola che rimuove la partizione specificata. La rimozione di una partizione altera solo i metadati della tabella esterna. I dati su Amazon S3 non sono interessati.

ROW LEVEL SECURITY { ON | OFF } [CONJUNCTION TYPE { AND | OR }] [FOR DATASHARES]

Una clausola che attiva o disattiva la sicurezza a livello di riga per una relazione.

Quando per una relazione è attivata la protezione a livello di riga, è possibile leggere solo le righe al livello di riga a cui la policy di sicurezza consente l'accesso. Quando non ci sono policy che consentono l'accesso alla relazione, non è possibile visualizzare alcuna riga dalla relazione. Solo gli utenti con privilegi avanzati e gli utenti o i ruoli che hanno il ruolo `sys:secadmin` possono impostare la clausola ROW LEVEL SECURITY. Per ulteriori informazioni, consulta Sicurezza a livello di riga.

- [CONJUNCTION TYPE { AND | OR }]

  Una clausola che consente di scegliere il tipo di congiunzione della policy di sicurezza a livello di riga per una relazione. Quando a una relazione sono associate più policy di sicurezza a livello di riga, è possibile combinare le policy con la clausola AND oppure OR. Per impostazione predefinita, Amazon Redshift combina le policy RLS con la clausola AND. Gli utenti con privilegi avanzati, gli utenti o i ruoli che hanno il ruolo `sys:secadmin` possono utilizzare questa clausola per definire il tipo di combinazione della policy di sicurezza a livello di riga per una relazione. Per ulteriori informazioni, consulta Combinazione di più policy per utente.

- FOR DATASHARES

  Una clausola che determina se è possibile accedere a una relazione protetta da RLS in un'unità di condivisione dati. Per impostazione predefinita, non è possibile accedere a una relazione protetta da RLS in un'unità di condivisione dati. Il comando ALTER TABLE ROW LEVEL SECURITY eseguito con questa clausola influisce solo sulla proprietà di accessibilità dell'unità di condivisione dati della relazione. La proprietà ROW LEVEL SECURITY non viene modificata.

  Se rendi accessibile una relazione protetta da RLS nelle unità di condivisione dati, la relazione non dispone di una sicurezza a livello di riga nel database con unità di condivisione dati sul lato consumer. La relazione mantiene la proprietà RLS sul lato producer.

Esempi

Per esempi che mostrano come utilizzare il comando ALTER TABLE, consultare quanto segue.

- Esempi di ALTER TABLE
- Esempi per alterare la tabella esterna
- Esempi di ALTER TABLE ADD e DROP COLUMN

Esempi di ALTER TABLE

I seguenti esempi dimostrano l'utilizzo di base del comando ALTER TABLE.

Ridenominazione di una tabella o una vista

Il seguente comando rinomina la tabella USERS in USERS_BKUP:

```
alter table users
rename to users_bkup;
```

Puoi anche usare questo tipo di comando per rinominare una vista.

Modifica del proprietario di una tabella o di una vista

Il seguente comando modifica il proprietario della tabella VENUE nell'utente DWUSER:

```
alter table venue
owner to dwuser;
```

I seguenti comandi creano una vista, quindi cambiano il suo proprietario:

```
create view vdate as select * from date;
alter table vdate owner to vuser;
```

Ridenominazione di una colonna

Il seguente comando rinomina la colonna VENUESEATS della tabella VENUE in VENUESIZE:

```
alter table venue
rename column venueseats to venuesize;
```

Rimozione del vincolo di una tabella

Per eliminare un vincolo di tabella, ad esempio una chiave primaria, una chiave esterna o un vincolo univoco, individua innanzitutto il nome interno del vincolo. Poi specifica il nome del vincolo nel comando ALTER TABLE. L'esempio seguente trova i vincoli per la tabella CATEGORY, quindi rimuove la chiave primaria con il nome `category_pkey`.

```
select constraint_name, constraint_type
from information_schema.table_constraints
where constraint_schema ='public'
and table_name = 'category';

constraint_name | constraint_type
----------------+----------------
category_pkey   | PRIMARY KEY

alter table category
drop constraint category_pkey;
```

Modificare una colonna VARCHAR

Per preservare lo storage, è possibile definire una tabella inizialmente con colonne VARCHAR con la dimensione minima necessaria per i requisiti di dati attuali. Successivamente, per utilizzare stringhe più lunghe, è possibile modificare la tabella per aumentare le dimensioni della colonna.

L'esempio seguente aumenta le dimensioni della colonna EVENTNAME in VARCHAR(300).

```
alter table event alter column eventname type varchar(300);
```

Modifica della codifica di compressione per una colonna

È possibile modificare la codifica di compressione di una colonna. Di seguito è riportata una serie di esempi che dimostrano questo approccio. La definizione di tabella per questi esempi è come segue.

```
create table t1(c0 int encode lzo, c1 bigint encode zstd, c2 varchar(16) encode lzo, c3
 varchar(32) encode zstd);
```

La seguente istruzione modifica la codifica di compressione per la colonna c0 dalla codifica LZO alla codifica AZ64.

```
alter table t1 alter column c0 encode az64;
```

La seguente istruzione modifica la codifica di compressione per la colonna c1 dalla codifica Zstandard alla codifica AZ64.

```
alter table t1 alter column c1 encode az64;
```

La seguente istruzione modifica la codifica di compressione per la colonna c2 dalla codifica LZO alla codifica Byte-Dictionary.

```
alter table t1 alter column c2 encode bytedict;
```

La seguente istruzione modifica la codifica di compressione per la colonna c3 dalla codifica Zstandard alla codifica Runlength.

```
alter table t1 alter column c3 encode runlength;
```

Modificare una colonna DISTSTYLE KEY DISTKEY

I seguenti esempi mostrano come modificare DISTSTYLE e DISTKEY di una tabella.

Crea una tabella con stile di distribuzione EVEN. La vista SVV_TABLE_INFO mostra che DISTSTYLE è EVEN.

```
create table inventory(
  inv_date_sk int4 not null ,
  inv_item_sk int4 not null ,
  inv_warehouse_sk int4 not null ,
  inv_quantity_on_hand int4
) diststyle even;

Insert into inventory values(1,1,1,1);

select "table", "diststyle" from svv_table_info;

   table   |   diststyle
-----------+----------------
 inventory |     EVEN
```

Modificare la tabella DISTKEY in `inv_warehouse_sk`. La vista SVV_TABLE_INFO mostra la colonna `inv_warehouse_sk` come chiave di distribuzione risultante.

```
alter table inventory alter diststyle key distkey inv_warehouse_sk;
```

```
select "table", "diststyle" from svv_table_info;

   table   |       diststyle
-----------+----------------------
 inventory | KEY(inv_warehouse_sk)
```

Modificare la tabella DISTKEY in `inv_item_sk`. La vista SVV_TABLE_INFO mostra la colonna `inv_item_sk` come chiave di distribuzione risultante.

```
alter table inventory alter distkey inv_item_sk;

select "table", "diststyle" from svv_table_info;

   table   |       diststyle
-----------+----------------------
 inventory | KEY(inv_item_sk)
```

Modificare una tabella in DISTSTYLE ALL

Gli esempi seguenti mostrano come modificare una tabella in DISTSTYLE ALL.

Crea una tabella con stile di distribuzione EVEN. La vista SVV_TABLE_INFO mostra che DISTSTYLE è EVEN.

```
create table inventory(
  inv_date_sk int4 not null ,
  inv_item_sk int4 not null ,
  inv_warehouse_sk int4 not null ,
  inv_quantity_on_hand int4
) diststyle even;

Insert into inventory values(1,1,1,1);

select "table", "diststyle" from svv_table_info;

   table   |   diststyle
-----------+---------------
 inventory |     EVEN
```

Modificare la tabella DISTSTYLE su ALL. La vista SVV_TABLE_INFO mostra il DISTSYTLE modificato.

```
alter table inventory alter diststyle all;

select "table", "diststyle" from svv_table_info;

   table   |   diststyle
-----------+----------------
 inventory |     ALL
```

Modifica una tabella SORTKEY

È possibile modificare una tabella per avere una chiave di ordinamento composta o nessuna chiave di ordinamento.

Nella seguente definizione della tabella, tabella `t1` è definita con una chiave di ordinamento interlacciata.

```
create table t1 (c0 int, c1 int) interleaved sortkey(c0, c1);
```

Il comando seguente modifica la tabella da una chiave di ordinamento interlacciata a una chiave di ordinamento composta.

```
alter table t1 alter sortkey(c0, c1);
```

Il comando seguente modifica la tabella per rimuovere la chiave di ordinamento interlacciata.

```
alter table t1 alter sortkey none;
```

Nella seguente definizione della tabella, tabella `t1` è definita con una colonna `c0` come chiave di ordinamento.

```
create table t1 (c0 int, c1 int) sortkey(c0);
```

Il comando seguente modifica la tabella `t1` in una chiave di ordinamento composta.

```
alter table t1 alter sortkey(c0, c1);
```

Modifica di una tabella in ENCODE AUTO

L'esempio seguente mostra come modificare una tabella in ENCODE AUTO.

La definizione di tabella per questo esempio è come segue. La colonna `c0` è definita con il tipo di codifica AZ64 mentre la colonna `c1` è definita con il tipo di codifica LZO.

```
create table t1(c0 int encode AZ64, c1 varchar encode LZO);
```

Per questa tabella, l'istruzione seguente modifica la codifica in AUTO.

```
alter table t1 alter encode auto;
```

L'esempio seguente mostra come modificare una tabella per rimuovere l'impostazione ENCODE AUTO.

La definizione di tabella per questo esempio è come segue. Le colonne della tabella sono definite senza codifica. In questo caso, la codifica predefinita è ENCODE AUTO.

```
create table t2(c0 int, c1 varchar);
```

Per questa tabella, la seguente istruzione modifica la codifica della colonna c0 in LZO. La codifica della tabella non è più impostata su ENCODE AUTO.

```
alter table t2 alter column c0 encode lzo;;
```

Modifica del controllo della sicurezza a livello di riga

Il seguente comando disattiva RLS per la tabella:

```
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY OFF;
```

Il seguente comando disattiva RLS (sicurezza a livello di riga) per la tabella:

```
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY ON;
```

Il seguente comando attiva RLS per la tabella e la rende accessibile nell'unità di condivisione dati:

```
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY ON;
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY FOR DATASHARES OFF;
```

Il seguente comando attiva RLS per la tabella e la rende inaccessibile nell'unità di condivisione dati:

```
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY ON;
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY FOR DATASHARES ON;
```

Il seguente comando attiva RLS e imposta il tipo di combinazione RLS su OR per la tabella:

```
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY ON CONJUNCTION TYPE OR;
```

Il seguente comando attiva RLS e imposta il tipo di combinazione RLS su AND per la tabella:

```
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY ON CONJUNCTION TYPE AND;
```

Esempi per alterare la tabella esterna

I seguenti esempi utilizzano un bucket Amazon S3 situato nella regione Stati Uniti orientali (Virginia settentrionale) (`us-east-1`) Regione AWS e le tabelle di esempio create in Esempi CREATE TABLE. Per ulteriori informazioni su come utilizzare le partizioni con tabelle esterne, consulta. Partizionamento delle tabelle esterne di Redshift Spectrum

L'esempio seguente imposta la proprietà della tabella numRows per la tabella esterna SPECTRUM.SALES su 170.000 righe.

```
alter table spectrum.sales
set table properties ('numRows'='170000');
```

L'esempio seguente modifica la posizione della tabella esterna SPECTRUM.SALES.

```
alter table spectrum.sales
set location 's3://redshift-downloads/tickit/spectrum/sales/';
```

L'esempio seguente modifica il formato della tabella esterna SPECTRUM.SALES in Parquet.

```
alter table spectrum.sales
set file format parquet;
```

L'esempio seguente aggiunge una partizione per la tabella SPECTRUM.SALES_PART.

```
alter table spectrum.sales_part
```

```
add if not exists partition(saledate='2008-01-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-01/';
```

L'esempio seguente aggiunge tre partizioni per la tabella SPECTRUM.SALES_PART.

```
alter table spectrum.sales_part add if not exists
partition(saledate='2008-01-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-01/'
partition(saledate='2008-02-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-02/'
partition(saledate='2008-03-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-03/';
```

L'esempio seguente altera SPECTRUM.SALES_PART per eliminare la partizione con `saledate='2008-01-01''`.

```
alter table spectrum.sales_part
drop partition(saledate='2008-01-01');
```

L'esempio seguente imposta un nuovo percorso Amazon S3 per la partizione con `saledate='2008-01-01'`.

```
alter table spectrum.sales_part
partition(saledate='2008-01-01')
set location 's3://redshift-downloads/tickit/spectrum/sales_partition/
saledate=2008-01-01/';
```

L'esempio seguente cambia il nome di `sales_date` in `transaction_date`.

```
alter table spectrum.sales rename column sales_date to transaction_date;
```

Il seguente esempio imposta la mappatura della colonna sulla mappatura in base alla posizione per una tabella esterna che utilizza il formato ORC (optimized row columnar).

```
alter table spectrum.orc_example
set table properties('orc.schema.resolution'='position');
```

Il seguente esempio imposta la mappatura della colonna sulla mappatura del nome per una tabella esterna che utilizza il formato ORC (optimized row columnar).

```
alter table spectrum.orc_example
set table properties('orc.schema.resolution'='name');
```

Esempi di ALTER TABLE ADD e DROP COLUMN

I seguenti esempi mostrano come usare ALTER TABLE per aggiungere e rimuovere una colonna di base e anche come eliminare una colonna con un oggetto dipendente.

ADD e DROP di una colonna di base

L'esempio seguente aggiunge una colonna autonoma FEEDBACK_SCORE alla tabella USERS. Questa colonna contiene semplicemente uninteger e il valore predefinito per questa colonna è NULL (nessun punteggio di feedback).

Per prima cosa, esegui una query sulla tabella di catalogo PG_TABLE_DEF per visualizzare lo schema della tabella USERS:

```
column        | type                    | encoding | distkey | sortkey
--------------+-------------------------+----------+---------+--------
userid        | integer                 | delta    | true    |       1
username      | character(8)            | lzo      | false   |       0
firstname     | character varying(30)   | text32k  | false   |       0
lastname      | character varying(30)   | text32k  | false   |       0
city          | character varying(30)   | text32k  | false   |       0
state         | character(2)            | bytedict | false   |       0
email         | character varying(100) | lzo      | false   |       0
phone         | character(14)           | lzo      | false   |       0
likesports    | boolean                 | none     | false   |       0
liketheatre   | boolean                 | none     | false   |       0
likeconcerts  | boolean                 | none     | false   |       0
likejazz      | boolean                 | none     | false   |       0
likeclassical | boolean                 | none     | false   |       0
likeopera     | boolean                 | none     | false   |       0
likerock      | boolean                 | none     | false   |       0
likevegas     | boolean                 | none     | false   |       0
likebroadway  | boolean                 | none     | false   |       0
likemusicals  | boolean                 | none     | false   |       0
```

Ora aggiungi la colonna feedback_score:

```
alter table users
add column feedback_score int
```

```
default NULL;
```

Seleziona la colonna FEEDBACK_SCORE da USERS per verificare che sia stata aggiunta:

```
select feedback_score from users limit 5;

feedback_score
----------------
NULL
NULL
NULL
NULL
NULL
```

Rimuovi la colonna per ripristinare la DDL originale:

```
alter table users drop column feedback_score;
```

Eliminazione di una colonna con un oggetto dipendente

Nell'esempio seguente viene rimossa una colonna che ha un oggetto dipendente. Di conseguenza, anche l'oggetto dipendente viene rimosso.

Per iniziare, aggiungi nuovamente la colonna FEEDBACK_SCORE alla tabella USERS:

```
alter table users
add column feedback_score int
default NULL;
```

Quindi, crea una vista dalla tabella USERS chiamata USERS_VIEW:

```
create view users_view as select * from users;
```

Ora prova a rimuovere la colonna FEEDBACK_SCORE dalla tabella USERS. Questa istruzione DROP utilizza il comportamento predefinito (RESTRICT):

```
alter table users drop column feedback_score;
```

Amazon Redshift visualizza un messaggio di errore indicante che la colonna non può essere rimossa perché un altro oggetto dipende da essa.

Prova a rimuovere nuovamente la colonna FEEDBACK_SCORE, questa volta specificando CASCADE per eliminare tutti gli oggetti dipendenti:

```
alter table users
drop column feedback_score cascade;
```

ALTER TABLE APPEND

Aggiunge le righe a una tabella di destinazione spostando i dati da una tabella di origine esistente. I dati nella tabella di origine vengono spostati nelle colonne corrispondenti della tabella di destinazione. L'ordine delle colonne non ha importanza. Dopo che i dati sono stati aggiunti alla tabella di destinazione, la tabella di origine è vuota. ALTER TABLE APPEND è in genere molto più veloce di un'operazione CREATE TABLE AS o INSERT INTO simile in quanto i dati vengono spostati, non duplicati.

> **Note**
>
> ALTER TABLE APPEND sposta i blocchi di dati tra la tabella di origine e la tabella di destinazione. Per migliorare le prestazioni, ALTER TABLE APPEND non compatta lo storage come parte dell'operazione di aggiunta. Di conseguenza, l'utilizzo della memoria aumenta temporaneamente. Per recuperare lo spazio, esegui un'operazione VACUUM.

Le colonne con lo stesso nome devono avere anche attributi di colonna identici. Se la tabella di origine o la tabella di destinazione contiene colonne che non esistono nell'altra tabella, utilizza i parametri IGNOREEXTRA o FILLTARGET per specificare la modalità di gestione delle colonne aggiuntive.

Non puoi aggiungere una colonna di identità. Se entrambe le tabelle includono una colonna di identità, il comando non riesce. Se solo una tabella ha una colonna di identità, includi il parametro FILLTARGET o IGNOREEXTRA. Per ulteriori informazioni, consulta Note per l'utilizzo di ALTER TABLE APPEND.

Puoi aggiungere una colonna GENERATED BY DEFAULT AS IDENTITY. Puoi aggiornare colonne definite come GENERATED BY DEFAULT AS IDENTITY con i valori forniti. Per ulteriori informazioni, consulta Note per l'utilizzo di ALTER TABLE APPEND.

La tabella di destinazione deve essere una tabella permanente. Tuttavia, l'origine può essere una tabella permanente o una vista materializzata configurata per l'importazione di dati in

streaming. Entrambe le tabelle devono utilizzare lo stesso stile di distribuzione e la stessa chiave di distribuzione, se ne è stata definita una. Se gli oggetti sono ordinati, entrambi gli oggetti devono utilizzare lo stesso stile di ordinamento e definire le stesse colonne come chiavi di ordinamento.

Un comando ALTER TABLE APPEND automaticamente esegue il commit al completamento dell'operazione. Non è possibile eseguire il rollback. Non è possibile eseguire ALTER TABLE APPEND all'interno di un blocco di transazioni (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Privilegi richiesti

A seconda del comando ALTER TABLE APPEND, è richiesto uno dei seguenti privilegi:

- Superuser
- Utenti con il privilegio di sistema ALTER TABLE
- Utenti con privilegi DELETE e SELECT sulla tabella di origine e privilegi INSERT sulla tabella di destinazione

Sintassi

```
ALTER TABLE target_table_name APPEND FROM [ source_table_name
 | source_materialized_view_name ]
[ IGNOREEXTRA | FILLTARGET ]
```

L'aggiunta da una vista materializzata funziona solo nel caso in cui la vista materializzata sia configurata per Importazione di dati in streaming.

Parametri

target_table_name

Il nome della tabella a cui vengono aggiunte le righe. Specifica solo il nome della tabella o utilizzare il formato schema_name.table_name per utilizzare uno schema specifico. La tabella di destinazione deve essere una tabella permanente esistente.

FROM source_table_name

Il nome della tabella che fornisce le righe da aggiungere. Specifica solo il nome della tabella o utilizzare il formato schema_name.table_name per utilizzare uno schema specifico. La tabella di origine deve essere una tabella permanente esistente.

FROM source_materialized_view_name

Il nome di una vista materializzata che fornisce le righe da aggiungere. L'aggiunta da una vista materializzata funziona solo nel caso in cui la vista materializzata sia configurata per Importazione di dati in streaming. La vista materializzata di origine deve esistere già.

IGNOREEXTRA

Parola chiave che specifica che se la tabella di origine include colonne che non sono presenti nella tabella di destinazione, i dati nelle colonne aggiuntive devono essere eliminati. Non puoi usare IGNOREEXTRA con FILLTARGET.

FILLTARGET

Parola chiave che specifica che se la tabella di origine include colonne che non sono presenti nella tabella di origine, le colonne devono contenere il valore di colonna DEFAULT se definito, altrimenti NULL. Non puoi usare IGNOREEXTRA con FILLTARGET.

Note per l'utilizzo di ALTER TABLE APPEND

ALTER TABLE APPEND sposta solo colonne identiche dalla tabella di origine alla tabella di destinazione. L'ordine delle colonne non ha importanza.

Se la tabella di origine o le tabelle di destinazione contengono colonne aggiuntive, utilizza FILLTARGET o IGNOREEXTRA in base alle seguenti regole:

- Se la tabella di origine contiene colonne che non esistono nella tabella di destinazione, includi IGNOREEXTRA. Il comando ignora le colonne aggiuntive nella tabella di origine.
- Se la tabella di destinazione contiene colonne che non esistono nella tabella di origine, includi FILLTARGET. Il comando inserisce nelle colonne aggiuntive della tabella di destinazione il valore di colonna predefinito o il valore IDENTITY se definito, altrimenti NULL.
- Se sia la tabella di origine che la tabella di destinazione contengono colonne aggiuntive, il comando non riesce. Non puoi usare entrambi FILLTARGET e IGNOREEXTRA.

Se una colonna con lo stesso nome ma attributi diversi esiste in entrambe le tabelle, il comando non riesce. Le colonne con nome simile devono avere i seguenti attributi in comune:

- Tipo di dati
- Dimensione colonna

- Codifica di compressione
- Non null
- Stile di ordinamento
- Colonne con chiave di ordinamento
- Stile di distribuzione
- Colonne con chiave di distribuzione

Non puoi aggiungere una colonna di identità. Se sia la tabella di origine che la tabella di destinazione hanno colonne di identità, il comando non riesce. Se solo la tabella di origine ha una colonna di identità, includi il parametro IGNOREEXTRA in modo che la colonna di identità venga ignorata. Se solo la tabella di destinazione ha una colonna Identity, includi il parametro FILLTARGET in modo che la colonna di identità venga compilata in base alla clausola IDENTITY definita per la tabella. Per ulteriori informazioni, consulta DEFAULT.

Puoi aggiungere una colonna di identità predefinita con l'istruzione ALTER TABLE APPEND. Per ulteriori informazioni, consulta CREATE TABLE.

Esempi di ALTER TABLE APPEND

Supponiamo che la tua organizzazione mantenga una tabella, SALES_MONTHLY, per acquisire le transazioni di vendita correnti. Vuoi spostare i dati dalla tabella delle transazioni alla tabella SALES, ogni mese.

Puoi utilizzare i seguenti comandi INSERT INTO e TRUNCATE per eseguire l'operazione.

```
insert into sales (select * from sales_monthly);
truncate sales_monthly;
```

Tuttavia, puoi eseguire la stessa operazione in modo molto più efficiente utilizzando il comando ALTER TABLE APPEND.

Innanzitutto, eseguire una query sulla tabella del catalogo di sistema PG_TABLE_DEF per verificare che entrambe le tabelle abbiano le stesse colonne con attributi di colonna identici.

```
select trim(tablename) as table, "column", trim(type) as type,
encoding, distkey, sortkey, "notnull"
from pg_table_def where tablename like 'sales%';
```

```
table      | column     | type                          | encoding | distkey | sortkey |
 notnull
-----------+------------+-------------------------------+----------+---------+---------
+--------
sales      | salesid    | integer                       | lzo      | false   |       0 |
 true
sales      | listid     | integer                       | none     | true    |       1 |
 true
sales      | sellerid   | integer                       | none     | false   |       2 |
 true
sales      | buyerid    | integer                       | lzo      | false   |       0 |
 true
sales      | eventid    | integer                       | mostly16 | false   |       0 |
 true
sales      | dateid     | smallint                      | lzo      | false   |       0 |
 true
sales      | qtysold    | smallint                      | mostly8  | false   |       0 |
 true
sales      | pricepaid  | numeric(8,2)                  | delta32k | false   |       0 |
 false
sales      | commission | numeric(8,2)                  | delta32k | false   |       0 |
 false
sales      | saletime   | timestamp without time zone | lzo      | false   |       0 |
 false
salesmonth | salesid    | integer                       | lzo      | false   |       0 |
 true
salesmonth | listid     | integer                       | none     | true    |       1 |
 true
salesmonth | sellerid   | integer                       | none     | false   |       2 |
 true
salesmonth | buyerid    | integer                       | lzo      | false   |       0 |
 true
salesmonth | eventid    | integer                       | mostly16 | false   |       0 |
 true
salesmonth | dateid     | smallint                      | lzo      | false   |       0 |
 true
salesmonth | qtysold    | smallint                      | mostly8  | false   |       0 |
 true
salesmonth | pricepaid  | numeric(8,2)                  | delta32k | false   |       0 |
 false
salesmonth | commission | numeric(8,2)                  | delta32k | false   |       0 |
 false
```

```
salesmonth | saletime   | timestamp without time zone | lzo      | false   |       0 |
 false
```

Quindi, guarda le dimensioni di ogni tabella.

```
select count(*) from sales_monthly;
 count
-------
  2000
(1 row)

select count(*) from sales;
 count
-------
 412,214
(1 row)
```

Ora esegui il seguente comando ALTER TABLE APPEND.

```
alter table sales append from sales_monthly;
```

Guarda di nuovo le dimensioni di ogni tabella. La tabella SALES_MONTHLY ora ha 0 righe e la tabella SALES è cresciuta di 2000 righe.

```
select count(*) from sales_monthly;
 count
-------
     0
(1 row)

select count(*) from sales;
 count
-------
 414214
(1 row)
```

Se la tabella di origine ha più colonne della tabella di destinazione, specifica il parametro IGNOREEXTRA. L'esempio seguente utilizza il parametro IGNOREEXTRA per ignorare le colonne aggiuntive nella tabella SALES_LISTING quando si accede alla tabella SALES.

```
alter table sales append from sales_listing ignoreextra;
```

Se la tabella di destinazione ha più colonne della tabella di origine, specifica il parametro FILLTARGET. L'esempio seguente utilizza il parametro FILLTARGET per popolare le colonne nella tabella SALES_REPORT che non esistono nella tabella SALES_MONTH.

```
alter table sales_report append from sales_month filltarget;
```

L'esempio seguente mostra un esempio di come utilizzare ALTER TABLE APPEND con una vista materializzata come origine.

```
ALTER TABLE target_tbl APPEND FROM my_streaming_materialized_view;
```

I nomi delle tabelle e delle viste materializzate in questo esempio sono esempi. L'aggiunta da una vista materializzata funziona solo nel caso in cui la vista materializzata sia configurata per Importazione di dati in streaming. Sposta tutti i record della vista materializzata di origine in una tabella di destinazione con lo stesso schema della vista materializzata e lascia intatta la vista materializzata. Si tratta dello stesso comportamento di quando l'origine dei dati è una tabella.

ALTER USER

Cambia un utente del database.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per ALTER USER:

- Superuser
- Utenti con il privilegio ALTER USER
- Utente attuale che desidera modificare la propria password

Sintassi

```
ALTER USER username [ WITH ] option [, ... ]

where option is

CREATEDB | NOCREATEDB
| CREATEUSER | NOCREATEUSER
```

```
| SYSLOG ACCESS { RESTRICTED | UNRESTRICTED }
| PASSWORD { 'password' | 'md5hash' | DISABLE }
[ VALID UNTIL 'expiration_date' ]
| RENAME TO new_name |
| CONNECTION LIMIT { limit | UNLIMITED }
| SESSION TIMEOUT limit | RESET SESSION TIMEOUT
| SET parameter { TO | = } { value | DEFAULT }
| RESET parameter
| EXTERNALID external_id
```

Parametri

username

Nome dell'utente.

WITH

Parola chiave facoltativa.

CREATEDB | NOCREATEDB

L'opzione CREATEDB consente all'utente di creare nuovi database. NOCREATEDB è il valore predefinito.

CREATEUSER | NOCREATEUSER

L'opzione CREATEUSER crea un utente con privilegi avanzati con tutti i privilegi del database, incluso CREATE USER. L'impostazione predefinita è NOCREATEUSER. Per ulteriori informazioni, consulta superuser.

SYSLOG ACCESS { RESTRICTED | UNRESTRICTED }

Una clausola che specifica il livello di accesso che l'utente ha per tabelle e viste di sistema di Amazon Redshift.

Gli utenti normali che dispongono dell'autorizzazione SYSLOG ACCESS RESTRICTED possono visualizzare solo le righe generate da quell'utente nelle tabelle e nelle viste di sistema visibili all'utente. Il valore predefinito è RESTRICTED.

Gli utenti normali che dispongono dell'autorizzazione SYSLOG ACCESS UNRESTRICTED possono visualizzare tutte le righe nelle tabelle e nelle viste di sistema visibili all'utente, incluse le righe generate da un altro utente. UNRESTRICTED non fornisce un accesso regolare all'utente

alle tabelle visibili agli utenti con privilegi avanzati. Solo gli utenti con privilegi avanzati possono vedere le tabelle visibili agli utenti con privilegi avanzati.

> **Note**
>
> Concedere all'utente un accesso illimitato alle tabelle di sistema offre all'utente la visibilità dei dati generati da altri utenti. Ad esempio, STL_QUERY e STL_QUERYTEXT contengono il testo completo delle istruzioni INSERT, UPDATE e DELETE, che potrebbero contenere dati sensibili generati dall'utente.

Tutte le righe in SVV_TRANSACTIONS sono visibili per tutti gli utenti.

Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

PASSWORD { 'password' | 'md5hash' | DISABLE }

Imposta la password dell'utente.

Per impostazione predefinita, gli utenti possono modificare le proprie password, a meno che la password non sia disabilitata. Per disabilitare la password di un utente, specifica DISABLE. Quando la password di un utente è disabilitata, la password viene eliminata dal sistema e l'utente può accedere solo utilizzando credenziali utente temporanee AWS Identity and Access Management (IAM). Per ulteriori informazioni, consultare Utilizzo dell'autenticazione IAM per generare credenziali utente di database. Solo un utente con privilegi avanzati può abilitare o disabilitare le password. Non puoi disabilitare la password di un utente con privilegi avanzati. Per abilitare una password, esegui ALTER USER e specifica una password.

Per i dettagli sull'utilizzo del parametro PASSWORD, consultare CREA UTENTE.

VALID UNTIL 'expiration_date'

Specifica che la password ha una data di scadenza. Usa il valore `'infinity'` per evitare di avere una data di scadenza. Il tipo di dati valido per questo parametro è il timestamp.

Solo gli utenti con privilegi avanzati possono utilizzare questo parametro.

RENAME TO

Rinomina l'utente.

new_name

Nuovo nome dell'utente. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

> **Important**
>
> Quando rinomini un utente, devi anche reimpostarne la password. La password reimpostata non deve necessariamente essere diversa dalla password precedente. Il nome utente viene utilizzato come parte della crittografia della password, quindi quando un utente viene rinominato, la password viene cancellata. L'utente non sarà in grado di accedere fino a quando la password viene ripristinata. Ad esempio:
>
> ```
> alter user newuser password 'EXAMPLENewPassword11';
> ```

CONNECTION LIMIT { limit | UNLIMITED }

Numero massimo di connessioni di database che l'utente può aprire contemporaneamente. Il limite non viene applicato per gli utente con privilegi avanzati. Utilizza la parola chiave UNLIMITED per consentire il numero massimo di connessioni simultanee. È possibile che venga applicato anche un limite al numero di connessioni per ciascun database. Per ulteriori informazioni, consulta CREATE DATABASE. Il valore predefinito è UNLIMITED. Per visualizzare le connessioni correnti, eseguire una query sulla vista di sistema STV_SESSIONS.

> **Note**
>
> Se si applicano entrambi i limiti di connessione utente e database, deve essere disponibile uno slot di connessione inutilizzato che rientra in entrambi i limiti quando un utente tenta di connettersi.

TIMEOUT SESSIONE limite | REIMPOSTARE IL TIMEOUT DELLA SESSIONE

Il tempo massimo (in secondi) in cui una sessione rimane inattiva o inutilizzata. L'intervallo è compreso tra 60 secondi (un minuto) e 1.728.000 secondi (20 giorni). Se per l'utente non è impostato alcun timeout di sessione, verrà applicata l'impostazione del cluster. Per ulteriori informazioni, consulta Quote e limiti in Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Quando si imposta il timeout della sessione, viene applicato solo alle nuove sessioni.

Per visualizzare informazioni sulle sessioni utente attive, tra cui l'ora di inizio, il nome utente e il timeout della sessione, eseguire una query sulla vista di sistema STV_SESSIONS. Per

visualizzare le informazioni sulla cronologia delle sessioni utente, eseguire una query sulla vista STL_SESSIONS. Per recuperare informazioni sugli utenti del database, inclusi i valori di timeout della sessione, eseguire una query sulla vista SVL_USER_INFO.

SET

Imposta un parametro di configurazione su un nuovo valore predefinito per tutte le sessioni eseguite dall'utente specificato.

RESET

Reimposta un parametro di configurazione sul valore predefinito originale per l'utente specificato.

parameter

Nome del parametro da impostare o reimpostare.

value

Nuovo valore del parametro.

DEFAULT

Imposta il parametro di configurazione sul valore predefinito per tutte le sessioni eseguite dall'utente specificato.

EXTERNALID external_id

L'identificativo dell'utente, associato a un provider di identità. L'utente deve avere la password disabilitata. Per ulteriori informazioni, consulta Native identity provider (IdP) federation for Amazon Redshift (Federazione di provider di identità nativi (IdP) per Amazon Redshift).

Note per l'utilizzo

- Tentativo di modificare rdsdb: non puoi modificare l'utente denominato `rdsdb`.
- Creazione di una password sconosciuta: quando si utilizza l'autenticazione AWS Identity and Access Management (IAM) per creare le credenziali utente del database, è possibile creare un superutente in grado di accedere solo utilizzando credenziali temporanee. Non puoi disabilitare la password di un utente con privilegi avanzati, ma puoi creare una password sconosciuta utilizzando una stringa di hash MD5 generata in modo casuale.

```
alter user iam_superuser password 'md512345678901234567801234567890l2';
```

- Impostazione di search_path: quando imposti il parametro search_path con il comando ALTER USER, la modifica diventa effettiva all'accesso successivo dell'utente specificato. Se si desidera modificare il valore di search_path per l'utente e la sessione correnti, utilizzare un comando SET.
- Impostazione del fuso orario: quando utilizzi SET TIMEZONE con il comando ALTER USER, la modifica diventa effettiva all'accesso successivo dell'utente specificato.
- Utilizzo del mascheramento dinamico dei dati e delle policy di sicurezza a livello di riga: quando il cluster o lo spazio dei nomi serverless di cui è stato effettuato il provisioning prevede il mascheramento dinamico dei dati o policy di sicurezza a livello di riga, i seguenti comandi sono bloccati per gli utenti normali:

  ```
  ALTER <current_user> SET enable_case_sensitive_super_attribute/
  enable_case_sensitive_identifier/downcase_delimited_identifier
  ```

  Solo gli utenti con privilegi avanzati e gli utenti con il privilegio ALTER USER possono impostare queste opzioni di configurazione. Per ulteriori informazioni sulla sicurezza a livello di riga. consulta Sicurezza a livello di riga. Per informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.

Esempi

L'esempio seguente concede all'utente ADMIN il privilegio di creare database:

```
alter user admin createdb;
```

L'esempio seguente imposta la password dell'utente ADMIN su `adminPass9` e imposta una data e un'ora di scadenza per la password:

```
alter user admin password 'adminPass9'
valid until '2017-12-31 23:59';
```

L'esempio seguente rinomina l'utente il gruppo ADMIN in SYSADMIN:

```
alter user admin rename to sysadmin;
```

Nell'esempio seguente il timeout della sessione di inattività per un utente viene aggiornato a 300 secondi.

```
ALTER USER dbuser SESSION TIMEOUT 300;
```

Reimposta il timeout della sessione di inattività dell'utente. Quando si reimposta, viene applicata l'impostazione del cluster. Per eseguire questo comando, è necessario essere un utente con privilegi avanzati del database. Per ulteriori informazioni, consulta Quote e limiti in Amazon Redshift nella Guida alla gestione di Amazon Redshift.

```
ALTER USER dbuser RESET SESSION TIMEOUT;
```

Nell'esempio seguente viene aggiornato l'ID esterno di un utente denominato bob. Lo spazio dei nomi è `myco_aad`. Se lo spazio dei nomi non è associato a un provider di identità registrato, si verifica un errore.

```
ALTER USER myco_aad:bob EXTERNALID "ABC123" PASSWORD DISABLE;
```

L'esempio seguente imposta il fuso orario per tutte le sessioni eseguite da un utente di database specifico. Modifica il fuso orario per le sessioni successive e non per quella corrente.

```
ALTER USER odie SET TIMEZONE TO 'Europe/Zurich';
```

L'esempio seguente imposta il numero massimo di connessioni al database che l'utente può bob tenere aperte.

```
ALTER USER bob CONNECTION LIMIT 10;
```

ANALYZE

Aggiorna le statistiche della tabella per l'utilizzo da parte del pianificatore di query.

Privilegi richiesti

Di seguito sono riportati i privilegi necessari per ANALYZE:

- Superuser
- Utenti con il privilegio ANALYZE
- Proprietario della relazione

- Proprietario del database con cui è condivisa la tabella

Sintassi

```
ANALYZE [ VERBOSE ]
[ [ table_name [ ( column_name [, ...] ) ] ]
[ PREDICATE COLUMNS | ALL  COLUMNS ]
```

Parametri

VERBOSE

Clausola che restituisce i messaggi di informazione sull'avanzamento relativi all'operazione ANALYZE. Questa opzione è utile quando non si specifica una tabella.

table_name

Puoi analizzare tabelle specifiche, incluse le tabelle temporanee. Puoi qualificare la tabella con il nome dello schema. Puoi specificare facoltativamente un table_name per analizzare una singola tabella. Non puoi specificare più di un table_name con una singola istruzione ANALYZE table_name. Se non si specifichi un valore nome_tabella, saranno analizzate tutte le tabelle nel database correntemente connesso, comprese le tabelle persistenti nel catalogo di sistema. Amazon Redshift salta l'analisi di una tabella se la percentuale di righe che sono state modificate dall'ultima analisi è inferiore alla soglia di analisi. Per ulteriori informazioni, consulta Soglia di analisi.

Non è necessario analizzare le tabelle di sistema di Amazon Redshift (tabelle STL e STV).

column_name

Se specifichi un table_name, puoi anche specificare una o più colonne nella tabella (come un elenco separato da colonne tra parentesi). Se viene specificato un elenco di colonne, vengono analizzate solo le colonne elencate.

PREDICATE COLUMNS | ALL COLUMNS

Clausole che indicano se ANALYZE deve includere solo le colonne di predicato. Specifica PREDICATE COLUMNS per analizzare solo le colonne che sono state utilizzate come predicati nelle query precedenti o sono probabili candidati da utilizzare come predicati. Specifica ALL COLUMNS per analizzare tutte le colonne. Il valore predefinito è ALL COLUMNS.

Una colonna è inclusa nel set di colonne dei predicati se una delle seguenti condizioni è vera:

- La colonna è stata utilizzata in una query come parte di un filtro, condizione di join o clausola group by.
- La colonna è una chiave di distribuzione.
- La colonna fa parte di una chiave di ordinamento.

Se nessuna colonna è contrassegnata come colonne di predicato, ad esempio perché la tabella non è stata ancora sottoposta a query, tutte le colonne vengono analizzate anche quando viene specificato PREDICATE COLUMNS. Per ulteriori informazioni sulle colonne di predicato, consultare Analisi delle tabelle.

Note per l'utilizzo

Amazon Redshift esegue automaticamente ANALYZE sulle tabelle create con i seguenti comandi:

- CREATE TABLE AS
- CREATE TEMP TABLE AS
- SELECT INTO

Non puoi analizzare una tabella esterna.

Non è necessario eseguire il comando ANALYZE su queste tabelle quando vengono create per la prima volta. Se le modifichi, devi analizzarle come fai per le altre tabelle.

Soglia di analisi

Per ridurre i tempi di elaborazione e migliorare le prestazioni generali del sistema, Amazon Redshift salta ANALYZE per una tabella se la percentuale di righe che sono state modificate dall'ultima esecuzione del comando ANALYZE è inferiore alla soglia di analisi specificata dal parametro analyze_threshold_percent. Per impostazione predefinita, `analyze_threshold_percent` è 10. Per cambiare `analyze_threshold_percent` per la sessione corrente, emettere il comando SET. L'esempio seguente cambia `analyze_threshold_percent` sul 20 percento.

```
set analyze_threshold_percent to 20;
```

Per analizzare le tabelle quando è stato modificato solo un piccolo numero di righe, imposta `analyze_threshold_percent` su un numero arbitrariamente piccolo. Ad esempio, se imposti `analyze_threshold_percent` su 0.01, una tabella con 100.000.000 di righe non viene ignorata se sono state modificate almeno 10.000 righe.

```
set analyze_threshold_percent to 0.01;
```

Se ANALYZE salta una tabella perché non soddisfa la soglia di analisi, Amazon Redshift restituisce il seguente messaggio.

```
ANALYZE SKIP
```

Per analizzare tutte le tabelle anche se non sono state modificate righe, imposta `analyze_threshold_percent` su 0.

Per visualizzare i risultati delle operazioni ANALYZE, eseguire una query della tabella di sistema STL_ANALYZE.

Per ulteriori informazioni sull'analisi delle tabelle, consultare Analisi delle tabelle.

Esempi

Analizza tutte le tabelle nel database TICKIT e restituisci le informazioni sull'avanzamento.

```
analyze verbose;
```

Analizza solo la tabella LISTING.

```
analyze listing;
```

Analizza le colonne VENUEID e VENUENAME nella tabella VENUE.

```
analyze venue(venueid, venuename);
```

Analizza solo le colonne di predicato della tabella VENUE.

```
analyze venue predicate columns;
```

ANALYZE COMPRESSION

Esegue l'analisi della compressione e produce un rapporto con la codifica di compressione suggerita per le tabelle analizzate. Per ogni colonna, il rapporto include una stima della potenziale riduzione dello spazio su disco rispetto alla codifica RAW.

Sintassi

```
ANALYZE COMPRESSION
[ [ table_name ]
[ ( column_name [, ...] ) ] ]
[COMPROWS numrows]
```

Parametri

table_name

Puoi analizzare la compressione di tabelle specifiche, incluse le tabelle temporanee. Puoi qualificare la tabella con il nome dello schema. Puoi specificare facoltativamente un table_name per analizzare una singola tabella. Se non specifichi un valore table_name, vengono analizzate tutte le tabelle nel database attualmente connesso. Non puoi specificare più di un table_name con una singola istruzione ANALYZE COMPRESSION.

column_name

Se specifichi un table_name, puoi anche specificare una o più colonne nella tabella (come un elenco separato da colonne tra parentesi).

COMPROWS

Numero di righe da utilizzare come dimensione del campione per l'analisi della compressione. L'analisi viene eseguita su righe da ciascuna sezione di dati. Ad esempio, se specifichi COMPROWS 1000000 (1.000.000) e il sistema contiene 4 sezioni totali, vengono lette e analizzate non più di 250.000 righe per sezione. Se COMPROWS non è specificato, la dimensione del campione è impostata su 100.000 per sezione. I valori di COMPROWS inferiori al valore predefinito di 100.000 righe per sezione vengono automaticamente aggiornati al valore predefinito. Tuttavia, l'analisi della compressione non produce raccomandazioni se la quantità di dati nella tabella è insufficiente per produrre un campione significativo. Se il numero COMPROWS è maggiore del numero di righe nella tabella, il comando ANALYZE COMPRESSION procede comunque ed esegue l'analisi della compressione su tutte le righe disponibili.

numrows

Numero di righe da utilizzare come dimensione del campione per l'analisi della compressione. L'intervallo accettato per numrows è un numero compreso tra 1000 e 1000000000 (1.000.000.000).

Note per l'utilizzo

ANALYZE COMPRESSION acquisisce un blocco di tabella esclusivo che impedisce letture e scritture simultanee nella tabella. Esegui il comando ANALYZE COMPRESSION solo quando la tabella è inattiva.

Esegui ANALYZE COMPRESSION per ottenere consigli per schemi di codifica delle colonne, basati su un campione dei contenuti della tabella. ANALYZE COMPRESSION è uno strumento di consulenza e non modifica le codifiche delle colonne della tabella. La codifica suggerita può essere applicata ricreando la tabella o creando una nuova tabella con lo stesso schema. Ricreare una tabella non compressa con schemi di codifica appropriati può ridurre significativamente il footprint su disco. Questo approccio consente di risparmiare spazio su disco e migliora le prestazioni delle query per i carichi di lavoro associati all'I/O.

ANALYZE COMPRESSION salta la fase di analisi effettiva e restituisce direttamente il tipo di codifica originale su qualsiasi colonna designata come SORTKEY. Lo fa perché le scansioni con limiti di intervallo potrebbero funzionare male quando le colonne SORTKEY sono molto più compresse di altre colonne.

Esempi

L'esempio seguente mostra la codifica e la riduzione percentuale stimata per le colonne solo nella tabella LISTING:

```
analyze compression listing;

  Table  |     Column     | Encoding | Est_reduction_pct
---------+----------------+----------+-------------------
 listing | listid         | az64     | 40.96
 listing | sellerid       | az64     | 46.92
 listing | eventid        | az64     | 53.37
 listing | dateid         | raw      | 0.00
 listing | numtickets     | az64     | 65.66
 listing | priceperticket | az64     | 72.94
 listing | totalprice     | az64     | 68.05
 listing | listtime       | az64     | 49.74
```

L'esempio seguente analizza le colonne QTYSOLD, COMMISSION e SALETIME nella tabella SALES.

```
analyze compression sales(qtysold, commission, saletime);
```

```
 Table |   Column   | Encoding | Est_reduction_pct
-------+------------+----------+-------------------
 sales | salesid    | N/A      | 0.00
 sales | listid     | N/A      | 0.00
 sales | sellerid   | N/A      | 0.00
 sales | buyerid    | N/A      | 0.00
 sales | eventid    | N/A      | 0.00
 sales | dateid     | N/A      | 0.00
 sales | qtysold    | az64     | 83.06
 sales | pricepaid  | N/A      | 0.00
 sales | commission | az64     | 71.85
 sales | saletime   | az64     | 49.63
```

ATTACH MASKING POLICY

Collega una policy di mascheramento dinamico dei dati esistente a una colonna. Per ulteriori informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.

Una policy di mascheramento può essere collegata da utenti con privilegi avanzati e da utenti o ruoli che dispongono del ruolo sys:secadmin.

Sintassi

```
ATTACH MASKING POLICY policy_name
   ON { relation_name }
   ( {output_columns_names | output_path} ) [ USING ( {input_column_names | input_path
 )} ]
   TO { user_name | ROLE role_name | PUBLIC }
   [ PRIORITY priority ];
```

Parametri

nome_policy

Nome della policy di mascheramento da collegare.

relation_name

Il nome della relazione a cui collegare la politica di mascheramento.

output_column_names

lNmi delle colonne a cui verrà applicata la policy di mascheramento.

output_paths

Il percorso completo dell'oggetto SUPER a cui viene applicata la politica di mascheramento, incluso il nome della colonna. Ad esempio, per una relazione con una colonna di tipo SUPER denominata `person`, output_path può essere `person.name.first_name`.

input_column_names

Nomi delle colonne che la policy di mascheramento utilizzerà come input. Questo parametro è facoltativo. Se non specificato, la politica di mascheramento utilizza come input output_column_names.

input_paths

Il percorso completo dell'oggetto SUPER che la politica di mascheramento utilizza come input. Questo parametro è facoltativo. Se non specificato, la politica di mascheramento utilizza come input output_path.

user_name

Nome dell'utente a cui verrà collegata la policy di mascheramento. Non è possibile collegare due politiche alla stessa combinazione di utente e colonna oppure ruolo e colonna. È possibile collegare una policy a un utente e un'altra policy al ruolo dell'utente. In questo caso, si applica la policy con la priorità più alta.

È possibile impostare solo uno tra user_name, role_name e PUBLIC in un singolo comando ATTACH MASKING POLICY.

role_name

Nome del ruolo a cui verrà collegata la policy di mascheramento. Non è possibile collegare due policy alla stessa coppia colonna/ruolo. È possibile collegare una policy a un utente e un'altra policy al ruolo dell'utente. In questo caso, si applica la policy con la priorità più alta.

È possibile impostare solo uno tra user_name, role_name e PUBLIC in un singolo comando ATTACH MASKING POLICY.

PUBLIC

Collega la policy di mascheramento a tutti gli utenti che accedono alla tabella. È necessario attribuire alle altre policy di mascheramento collegate a specifiche coppie colonna/utente o colonna/ruolo una priorità maggiore rispetto alla policy PUBLIC affinché possano essere applicate.

È possibile impostare solo uno tra user_name, role_name e PUBLIC in un singolo comando ATTACH MASKING POLICY.

priority

La priorità della policy di mascheramento. Quando si applicano più policy di mascheramento alla query di un determinato utente, si applica la policy con la massima priorità.

Non è possibile collegare due politiche diverse alla stessa colonna con la medesima priorità, anche se sono collegate a utenti o ruoli diversi. È possibile collegare la stessa politica più volte allo stesso set di tabella, colonna di output, colonna di input e parametri di priorità, purché l'utente o il ruolo a cui si riferisce la politica ogni volta sia diverso.

Non è possibile applicare una policy a una colonna con la stessa priorità di un'altra policy associato a quella colonna, anche se si tratta di ruoli diversi. Questo campo è facoltativo. Se non si specifica una priorità, per impostazione predefinita la policy di mascheramento prevede l'associazione con una priorità pari a 0.

ATTACH RLS POLICY

Collegamento di una policy di sicurezza a livello di riga su una tabella a uno o più utenti o ruoli.

Una policy può essere collegata da superuser e utenti o ruoli che dispongono del ruolo `sys:secadmin`.

Sintassi

```
ATTACH RLS POLICY policy_name ON [TABLE] table_name [, ...]
TO { user_name | ROLE role_name | PUBLIC } [, ...]
```

Parametri

nome_policy

Il nome della policy .

ON [TABLE] nome_tabella [, ...]

La relazione a cui è collegata la policy di sicurezza a livello di riga.

TO { nome_utente | ROLE nome_ruolo | PUBLIC} [, ...]

Specifica se la policy è collegata a uno o più utenti o ruoli specificati.

Note per l'utilizzo

Quando lavori con l'istruzione ATTACH RLS POLICY, tieni presente quanto segue:

- La tabella che viene collegata deve contenere tutte le colonne elencate nella clausola WITH dell'istruzione di creazione della policy.
- Amazon Redshift RLS non supporta il collegamento di policy RLS ai seguenti oggetti:
  - Tabelle del catalogo
  - Relazioni tra database
  - Tabelle esterna
  - Viste materializzate
  - Tabelle temporanee
  - Tabelle di ricerca
- Non è possibile collegare una policy RLS a utenti con privilegi avanzati o a utenti con l'autorizzazione `sys:secadmin`.

Esempi

Nell'esempio seguente viene collegata una policy su una tabella a un ruolo.

```
ATTACH RLS POLICY policy_concerts ON tickit_category_redshift TO ROLE analyst, ROLE
 dbadmin;
```

BEGIN

Inizia una transazione. Sinonimo di START TRANSACTION.

Una transazione è una singola unità logica di lavoro composta da uno o più comandi. In generale, tutti i comandi in una transazione vengono eseguiti in uno snapshot del database la cui ora di inizio è determinata dal valore impostato per il parametro di configurazione del sistema `transaction_snapshot_begin`.

Per impostazione predefinita, le singole operazioni Amazon Redshift (query, istruzioni DDL, caricamenti) vengono sottoposte automaticamente al commit nel database. Se vuoi sospendere il commit per un'operazione fino al completamento del lavoro successivo, è necessario aprire una transazione con l'istruzione BEGIN, quindi eseguire i comandi richiesti e chiudere la transazione con un'istruzione COMMIT o END. Se necessario, è possibile utilizzare un'istruzione ROLLBACK

per interrompere una transazione in corso. Un'eccezione a questo comportamento è costituita dal comando TRUNCATE che esegue il commit della transazione in cui viene eseguito e non può essere sottoposto al rollback.

Sintassi

```
BEGIN [ WORK | TRANSACTION ] [ ISOLATION LEVEL option ] [ READ WRITE | READ ONLY ]

START TRANSACTION [ ISOLATION LEVEL option ] [ READ WRITE | READ ONLY ]

Where option is

SERIALIZABLE
| READ UNCOMMITTED
| READ COMMITTED
| REPEATABLE READ

Note: READ UNCOMMITTED, READ COMMITTED, and REPEATABLE READ have no
operational impact and map to SERIALIZABLE in Amazon Redshift. You can see database
 isolation levels on your cluster
by querying the stv_db_isolation_level table.
```

Parametri

WORK

Parola chiave facoltativa.

TRANSACTION

Parola chiave facoltativa; WORK e TRANSACTION sono sinonimi.

ISOLATION LEVEL SERIALIZABLE

L'isolamento serializzabile è supportato per impostazione predefinita, quindi il comportamento della transazione è lo stesso indipendentemente dal fatto che questa sintassi sia inclusa nell'istruzione. Per ulteriori informazioni, consulta Gestione delle operazioni di scrittura simultanee. Non sono supportate altri livelli di isolamento.

> **Note**
>
> Lo standard SQL definisce quattro livelli di isolamento della transazione per impedire letture modificate (in cui una transazione legge i dati scritti da una transazione concorrente

non impegnata), letture non ripetibili (in cui una transazione rilegge i dati letti in precedenza e trova che i dati sono stati modificati da un'altra transazione che ha eseguito il commit dalla lettura iniziale) e letture fantasma (in cui una transazione esegue nuovamente una query, restituisce un set di righe che soddisfa una condizione di ricerca e poi trova che il set di righe è cambiato a causa di un'altra transazione recentemente sottoposta al commit):

- Lettura non sottoposta al commit: sono possibili letture modificate, letture non ripetibili e letture fantasma.
- Lettura sottoposta al commit: sono possibili letture non ripetibili e letture fantasma.
- Lettura ripetibile: sono possibili letture fantasma.
- Serializzabile: impedisce letture modificate, letture non ripetibili e letture fantasma.

Sebbene sia possibile utilizzare uno qualsiasi dei quattro livelli di isolamento della transazione, Amazon Redshift elabora tutti i livelli di isolamento come serializzabili.

READ WRITE

Fornisce le autorizzazioni di lettura e scrittura alla transazione.

READ ONLY

Fornisce solo le autorizzazioni di lettura alla transazione.

Esempi

L'esempio seguente avvia un blocco di transazione serializzabile:

```
begin;
```

L'esempio seguente avvia il blocco di transazione con un livello di isolamento serializzabile e le autorizzazioni di lettura e scrittura:

```
begin read write;
```

CALL

Esegue una procedura archiviata. Il comando CALL deve includere il nome della procedura e i valori dell'argomento di input. Per chiamare una procedura archiviata, utilizza l'istruzione CALL.

> **Note**
>
> CALL non può fare parte di alcuna query normale.

Sintassi

```
CALL sp_name ( [ argument ] [, ...] )
```

Parametri

sp_name

Il nome della procedura da eseguire.

argument

Il valore dell'argomento di input. Questo parametro può essere anche un nome di funzione, ad esempio `pg_last_query_id()`. Non puoi utilizzare le query come argomenti CALL.

Note per l'utilizzo

Le procedure archiviate di Amazon Redshift supportano le chiamate nidificate e ricorsive, come descritto di seguito. Inoltre, assicurati che il supporto del driver sia up-to-date, come descritto di seguito.

Argomenti

- Chiamate nidificate
- Supporto driver

Chiamate nidificate

Le procedure archiviate di Amazon Redshift supportano le chiamate nidificate e ricorsive. Il numero massimo consentito di livelli di nidificazione è 16. Le chiamate nidificate possono incapsulare la logica di business in procedure più piccole, condivisibili da più intermediari.

Se chiami una procedura nidificata che dispone di parametri di output, la procedura interna deve definire argomenti INOUT. In questo caso, la procedura interna viene trasmessa a una variabile non

costante. Gli argomenti OUT non sono consentiti. Questo comportamento si verifica in quanto una variabile è necessaria per mantenere l'output di una chiamata interna.

La relazione tra procedure interne ed esterne è registrata nella colonna `from_sp_call` di SVL_STORED_PROC_CALL.

L'esempio seguente mostra la trasmissione di variabili a una chiamata di procedura nidificata tramite argomenti INOUT.

```
CREATE OR REPLACE PROCEDURE inner_proc(INOUT a int, b int, INOUT c int) LANGUAGE
 plpgsql
AS $$
BEGIN
  a := b * a;
  c := b * c;
END;
$$;

CREATE OR REPLACE PROCEDURE outer_proc(multiplier int) LANGUAGE plpgsql
AS $$
DECLARE
  x int := 3;
  y int := 4;
BEGIN
  DROP TABLE IF EXISTS test_tbl;
  CREATE TEMP TABLE test_tbl(a int, b varchar(256));
  CALL inner_proc(x, multiplier, y);
  insert into test_tbl values (x, y::varchar);
END;
$$;

CALL outer_proc(5);

SELECT * from test_tbl;
 a  | b
----+----
 15 | 20
(1 row)
```

Supporto driver

È consigliabile aggiornare i driver Java Database Connectivity (JDBC) e Open Database Connectivity (ODBC) alla versione più recente che dispone del supporto per le procedure archiviate di Amazon Redshift.

Se lo strumento del tuo client utilizza le operazioni API che trasmettono al server tramite l'istruzione CALL, devi essere in grado di utilizzare il driver esistente. Se esistenti, i parametri di output vengono restituiti come insieme di risultati di una riga.

Le versioni più recenti dei driver JDBC e ODBC di Amazon Redshift dispongono del supporto dei metadati per il rilevamento delle procedure archiviate. oltre che del supporto `CallableStatement` per le applicazioni Java personalizzate. Per ulteriori informazioni sui driver, consulta Connessione a un cluster Amazon Redshift tramite gli strumenti del client SQL nella Guida alla gestione di Amazon Redshift.

Gli esempi seguenti mostrano come utilizzare operazioni API diverse del driver JDBC per le chiamate delle procedure archiviate.

```
void statement_example(Connection conn) throws SQLException {
  statement.execute("CALL sp_statement_example(1)");
}

void prepared_statement_example(Connection conn) throws SQLException {
  String sql = "CALL sp_prepared_statement_example(42, 84)";
  PreparedStatement pstmt = conn.prepareStatement(sql);
  pstmt.execute();
}

void callable_statement_example(Connection conn) throws SQLException {
  CallableStatement cstmt = conn.prepareCall("CALL sp_create_out_in(?,?)");
  cstmt.registerOutParameter(1, java.sql.Types.INTEGER);
  cstmt.setInt(2, 42);
  cstmt.executeQuery();
  Integer out_value = cstmt.getInt(1);
}
```

Esempi

L'esempio seguente chiama il nome della procedura `test_spl`.

```
call test_sp1(3,'book');
```

```
INFO:  Table "tmp_tbl" does not exist and will be skipped
INFO:  min_val = 3, f2 = book
```

L'esempio seguente chiama il nome della procedura `test_spl2`.

```
call test_sp2(2,'2019');

         f2          | column2
---------------------+---------
 2019+2019+2019+2019 | 2
(1 row)
```

CANCEL

Annulla una query del database attualmente in esecuzione.

Il comando CANCEL richiede l'ID di processo o l'ID di sessione della query in esecuzione e visualizza un messaggio di conferma per verificare che la query sia stata annullata.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per CANCEL:

- Utente con privilegi avanzati che annulla la propria query
- Utente con privilegi avanzati che annulla la query di un utente
- Utenti con il privilegio CANCEL che annulla la query di un utente
- Utente che annulla la propria query

Sintassi

```
CANCEL process_id [ 'message' ]
```

Parametri

process_id

Per annullare una query in esecuzione in un cluster Amazon Redshift, utilizza il `pid` (Process ID) corrispondente alla query che desideri annullare. STV_RECENTS

Per annullare una query in esecuzione in un gruppo di lavoro Serverless Amazon Redshift, utilizza il `session_id` modulo SYS_QUERY_HISTORY corrispondente alla query che desideri annullare.

'message'

Messaggio di conferma facoltativo che viene visualizzato al termine dell'annullamento della query. Se non si specifica un messaggio, Amazon Redshift visualizza il messaggio predefinito come verifica. È necessario racchiudere il messaggio tra virgolette singole.

Note per l'utilizzo

Non puoi annullare una query specificando un ID di query; devi specificare l'ID di processo (PID) o l'ID di sessione della query. Puoi annullare solo le query attualmente eseguite dall'utente. Gli utenti con privilegi avanzati possono annullare tutte le query.

Se le query in più sessioni contengono blocchi sulla stessa tabella, è possibile utilizzare la funzione PG_TERMINATE_BACKEND per terminare una delle sessioni. Questa operazione forza qualsiasi transazione attualmente in esecuzione nella sessione terminata al fine di rilasciare tutti i blocchi ed eseguire il rollback della transazione. Per visualizzare i blocchi correnti, eseguire una query sulla tabella di sistema STV_LOCKS.

Seguendo determinati eventi interni, Amazon Redshift può riavviare una sessione attiva e assegnare un nuovo PID. Se il PID è cambiato, è possibile che venga ricevuto il seguente messaggio di errore:

```
Session <PID> does not exist. The session PID might have changed. Check the
 stl_restarted_sessions system table for details.
```

Per trovare il nuovo PID, eseguire una query sulla tabella di sistema STL_RESTARTED_SESSIONS e filtrare la colonna `oldpid`.

```
select oldpid, newpid from stl_restarted_sessions where oldpid = 1234;
```

Esempi

Per annullare una query attualmente in esecuzione in un cluster Amazon Redshift, recupera innanzitutto l'ID del processo per la query che desideri annullare. Per determinare gli ID processo di tutte le query attualmente in esecuzione, digita il seguente comando:

```
select pid, starttime, duration,
```

```
trim(user_name) as user,
trim (query) as querytxt
from stv_recents
where status = 'Running';

pid |         starttime          | duration |   user   |    querytxt
-----+----------------------------+----------+----------+-----------------
802 | 2008-10-14 09:19:03.550885 |      132 | dwuser | select
venuename from venue where venuestate='FL', where venuecity not in
('Miami' , 'Orlando');
834 | 2008-10-14 08:33:49.473585 |  1250414 | dwuser | select *
from listing;
964 | 2008-10-14 08:30:43.290527 |   326179 | dwuser | select
sellerid from sales where qtysold in (8, 10);
```

Controlla il testo della query per determinare quale ID processo (PID) corrisponde alla query che vuoi annullare.

Digita il seguente comando per utilizzare PID 802 per annullare quella query:

```
cancel 802;
```

La sessione in cui è stata eseguita la query visualizza il seguente messaggio:

```
ERROR:  Query (168) cancelled on user's request
```

dove 168 è l'ID query (non l'ID processo utilizzato per annullare la query).

In alternativa, puoi specificare un messaggio di conferma personalizzato da visualizzare al posto del messaggio predefinito. Per specificare un messaggio personalizzato, includi il messaggio tra virgolette singole alla fine del comando CANCEL:

```
cancel 802 'Long-running query';
```

La sessione in cui è stata eseguita la query visualizza il seguente messaggio:

```
ERROR:  Long-running query
```

CLOSE

(Facoltativo) Chiude tutte le risorse libere associate a un cursore aperto. COMMIT, END e ROLLBACK chiudono automaticamente il cursore, quindi non è necessario usare il comando CLOSE per chiudere esplicitamente il cursore.

Per ulteriori informazioni, consultare DECLARE, FETCH.

Sintassi

```
CLOSE cursor
```

Parametri

cursor

Nome del cursore da chiudere.

Esempio di CLOSE

I seguenti comandi chiudono il cursore ed eseguono un commit che termina la transazione:

```
close movie_cursor;
commit;
```

COMMENT

Crea o modifica un commento su un oggetto di database.

Sintassi

```
COMMENT ON
{
TABLE object_name |
COLUMN object_name.column_name |
CONSTRAINT constraint_name ON table_name |
DATABASE object_name |
VIEW object_name
}
IS 'text' | NULL
```

Parametri

object_name

Nome dell'oggetto del database che viene commentato. Puoi aggiungere un commento ai seguenti oggetti:

- TABLE
- COLUMN (prende anche un column_name).
- CONSTRAINT (prende anche un constraint_name e un table_name).
- DATABASE
- VIEW
- SCHEMA

IS 'text' | NULL

Il testo del commento che si desidera aggiungere o sostituire per l'oggetto specificato. La stringa text ha un tipo di dati TEXT. Racchiudi il commento tra virgolette singole. Impostare il valore su NULL per rimuovere il testo del commento.

column_name

Nome della colonna che viene commentata. Parametro di COLUMN. Segue una tabella specificata in `object_name`.

constraint_name

Nome del vincolo che viene commentato. Parametro di CONSTRAINT.

table_name

Nome della tabella contenente il vincolo. Parametro di CONSTRAINT.

Note per l'utilizzo

Per aggiungere o aggiornare un commento, si deve essere un superuser o il proprietario di un oggetto di database.

I commenti sui database possono essere applicati solo al database corrente. Viene visualizzato un messaggio di avviso se tenti di commentare un database diverso. Lo stesso avviso viene visualizzato per i commenti su database che non esistono.

I commenti su tabelle esterne, colonne esterne e colonne di viste con associazione tardiva non sono supportati.

Esempi

Nell'esempio seguente viene aggiunto un commento alla tabella SALES.

```
COMMENT ON TABLE sales IS 'This table stores tickets sales data';
```

Nell'esempio seguente viene visualizzato il commento nella tabella SALES.

```
select obj_description('public.sales'::regclass);

obj_description
------------------------------------
This table stores tickets sales data
```

Nell'esempio seguente viene rimosso un commento dalla tabella SALES.

```
COMMENT ON TABLE sales IS NULL;
```

Nell'esempio seguente viene aggiunto un commento alla colonna EVENTID della tabella SALES.

```
COMMENT ON COLUMN sales.eventid IS 'Foreign-key reference to the EVENT table.';
```

Nell'esempio seguente viene visualizzato un commento nella colonna EVENTID (colonna numero 5) della tabella SALES.

```
select col_description( 'public.sales'::regclass, 5::integer );

col_description
----------------------------------------
Foreign-key reference to the EVENT table.
```

Nell'esempio seguente viene aggiunto un commento descrittivo alla tabella EVENT.

```
comment on table event is 'Contains listings of individual events.';
```

Per visualizzare i commenti, eseguire una query sul catalogo di sistema PG_DESCRIPTION. L'esempio seguente restituisce la descrizione per la tabella EVENT.

```
select * from pg_catalog.pg_description
where objoid =
(select oid from pg_class where relname = 'event'
and relnamespace =
(select oid from pg_catalog.pg_namespace where nspname = 'public') );

objoid | classoid | objsubid | description
-------+----------+----------+---------------------------------------
116658 |     1259 |        0 | Contains listings of individual events.
```

COMMIT

Esegue il commit della transazione corrente nel database. Questo comando rende permanenti gli aggiornamenti del database dalla transazione.

Sintassi

```
COMMIT [ WORK | TRANSACTION ]
```

Parametri

WORK

Parola chiave facoltativa. Questa parola chiave non è supportata all'interno di una procedura archiviata.

TRANSACTION

Parola chiave facoltativa. WORK e TRANSACTION sono sinonimi. Nessuno dei due è supportato all'interno d una procedura archiviata.

Per informazioni su come utilizzare COMMIT all'interno di una procedura archiviata, consultare Gestione delle transazioni.

Esempi

Ciascuno degli esempi seguenti esegue il commit della transazione corrente nel database:

```
commit;
```

```
commit work;
```

```
commit transaction;
```

COPY

Carica i dati in una tabella dai file di dati o da una tabella Amazon DynamoDB. I file possono trovarsi in un bucket Amazon Simple Storage Service (Amazon S3), un cluster Amazon EMR o un host remoto che accede tramite una connessione Secure Shell (SSH).

> **Note**
>
> Le tabelle esterne di Amazon Redshift Spectrum sono di sola lettura. Non è possibile eseguire il comando COPY per una tabella esterna.

Il comando COPY aggiunge i nuovi dati di input come righe aggiuntive nella tabella.

La dimensione massima di una singola riga di input da qualsiasi origine è pari a 4 MB.

Argomenti

Autorizzazioni richieste

Per utilizzare il comando COPY, è necessario disporre del privilegio INSERT per la tabella Amazon Redshift.

Sintassi di COPY

```
COPY table-name
[ column-list ]
FROM data_source
authorization
[ [ FORMAT ] [ AS ] data_format ]
[ parameter [ argument ] [, ... ] ]
```

È possibile eseguire un'operazione COPY con soli tre parametri: un nome di tabella, un'origine dati e l'autorizzazione per accedere ai dati.

Amazon Redshift estende la funzionalità del comando COPY per consentirti di caricare i dati in diversi formati di dati da diverse origini dati, controllare l'accesso per caricare i dati, gestire le trasformazioni di dati e gestire l'operazione di caricamento.

Le seguenti sezioni presentano i parametri obbligatori del comando COPY e raggruppano i parametri opzionali per funzione. Gli argomenti successivi descrivono ogni parametro e spiegano come interagiscono le diverse opzioni. Puoi anche andare direttamente alla descrizione di un parametro utilizzando un elenco alfabetico di parametri.

Parametri obbligatori

Il comando COPY necessita di tre elementi:

- Table Name
- Data Source
- Authorization

Il comando COPY più semplice utilizza il seguente formato.

```
COPY table-name
FROM data-source
```

```
authorization;
```

L'esempio seguente crea una tabella chiamata CATDEMO, poi carica la tabella con dati campione da un file di dati in Amazon S3 chiamato `category_pipe.txt`.

```
create table catdemo(catid smallint, catgroup varchar(10), catname varchar(10), catdesc
 varchar(50));
```

Nell'esempio seguente, l'origine dati per il comando COPY è un file di dati chiamato `category_pipe.txt` nella cartella `tickit` di un bucket Amazon S3 chiamato `redshift-downloads`. Il comando COPY è autorizzato ad accedere al bucket Amazon S3 tramite un ruolo AWS Identity and Access Management (IAM). Se il cluster possiede un ruolo IAM esistente per accedere ad Amazon S3 collegato, è possibile sostituire l'Amazon Resource Name (ARN) del ruolo nel comando COPY successivo ed eseguirlo.

```
copy catdemo
from 's3://redshift-downloads/tickit/category_pipe.txt'
iam_role 'arn:aws:iam::<aws-account-id>:role/<role-name>'
region 'us-east-1';
```

Per istruzioni complete su come utilizzare i comandi COPY per caricare dati di esempio, incluse istruzioni per caricare dati da altre AWS regioni, consulta Load Sample Data from Amazon S3 nella Amazon Redshift Getting Started Guide.

table-name

Il nome della tabella di destinazione per il comando COPY. La tabella deve esistere già nel database. La tabella può essere temporanea o persistente. Il comando COPY aggiunge i nuovi dati di input a tutte le righe esistenti nella tabella.

FROM data-source

La posizione dei dati di origine da caricare nella tabella di destinazione. È possibile specificare un file manifest con alcune origini dati.

Di solito, il repository dei dati più utilizzato è un bucket Amazon S3. È possibile anche caricare da file di dati ubicati in un cluster Amazon EMR, un'istanza Amazon EC2 o un host remoto a cui il cluster può accedere utilizzando una connessione SSH oppure è possibile caricare direttamente da una tabella DynamoDB.

- COPY da Amazon S3

- COPY da Amazon EMR
- COPY da host remoto (SSH)
- COPY da Amazon DynamoDB

Autorizzazione

Una clausola che indica il metodo utilizzato dal cluster per l'autenticazione e l'autorizzazione all'accesso ad altre risorse. AWS Il comando COPY richiede l'autorizzazione per accedere ai dati in un'altra AWS risorsa, tra cui Amazon S3, Amazon EMR, Amazon DynamoDB e Amazon EC2. È possibile fornire tale autorizzazione referenziando il ruolo IAM collegato al cluster o fornendo l'ID chiave di accesso e la chiave di accesso segreta per un utente IAM.

- Parametri di autorizzazione
- Controllo degli accessi basato sui ruoli
- Controllo degli accessi basato su chiave

Parametri facoltativi

È possibile specificare facoltativamente il modo in cui COPY mappa i dati dei campi in colonne nella tabella di destinazione, definire gli attributi dei dati di origine per consentire al comando COPY di leggere correttamente e analizzare i dati di origine e gestire le operazioni che il comando COPY deve eseguire durante il processo di caricamento.

- Opzioni di mappatura di colonne
- Parametri del formato dei dati
- Parametri di conversione dei dati
- Operazioni di caricamento dati

Mappatura di colonne

Per impostazione predefinita, COPY inserisce i valori dei campi nelle colonne della tabella di destinazione nello stesso ordine in cui i campi si presentano nei file di dati. Se l'ordine predefinito delle colonne non funziona, è possibile specificare un elenco di colonne o utilizzare delle espressioni JSONPath per mappare i campi dei dati di origine nelle colonne di destinazione.

- Column List
- JSONPaths File

Parametri del formato dei dati

È possibile caricare dati da file di testo in formato larghezza fissa, delimitato da caratteri, valori separati da virgole (CSV) o JSON o da file Avro.

Per impostazione predefinita, il comando COPY prevede che i dati di origine siano in file di testo UTF-8 delimitati da caratteri. Il delimitatore predefinito è una barra verticale (|). Se i dati di origine sono in un altro formato, utilizza i seguenti parametri per specificare il formato dei dati.

- FORMAT
- CSV
- DELIMITER
- FIXEDWIDTH
- SHAPEFILE
- AVRO
- JSON
- ENCRYPTED
- BZIP2
- GZIP
- LZOP
- PARQUET
- ORC
- ZSTD

Parametri di conversione dei dati

Mentre carica la tabella, COPY tenta di convertire in modo implicito le stringhe nei dati di origine nel tipo di dati della colonna di destinazione. Se hai necessità di specificare una conversione diversa dal comportamento predefinito o se la conversione predefinita dà luogo a errori, è possibile gestire le conversioni dei dati specificando i seguenti parametri.

- ACCEPTANYDATE
- ACCEPTINVCHARS
- BLANKSASNULL
- DATEFORMAT

- EMPTYASNULL
- ENCODING
- ESCAPE
- EXPLICIT_IDS
- FILLRECORD
- IGNOREBLANKLINES
- IGNOREHEADER
- NULL AS
- REMOVEQUOTES
- ROUNDEC
- TIMEFORMAT
- TRIMBLANKS
- TRUNCATECOLUMNS

Operazioni di caricamento dati

Gestisce il comportamento predefinito dell'operazione di caricamento per la risoluzione dei problemi o per ridurre i tempi di caricamento specificando i seguenti parametri.

- COMPROWS
- COMPUPDATE
- IGNOREALLERRORS
- MAXERROR
- NOLOAD
- STATUPDATE

Note sull'utilizzo e risorse aggiuntive per il comando COPY

Per ulteriori informazioni su come utilizzare il comando COPY, consultare i seguenti argomenti:

- Note per l'utilizzo
- Tutorial: Caricamento dei dati da Amazon S3
- Best practice di Amazon Redshift per il caricamento di dati

- Utilizzo di un comando COPY per il caricamento dei dati
  - Caricamento di dati da Amazon S3
  - Caricamento di dati da Amazon EMR
  - Caricamento di dati da host remoti
  - Caricamento di dati da una tabella Amazon DynamoDB
- Risoluzione di problemi di caricamento dei dati

Esempi di comando COPY

Per altri esempi che mostrano come eseguire COPY da varie origini, in formati diversi e con diverse opzioni di COPY, consulta Esempi di COPY.

COPY JOB (anteprima)

Questa è la documentazione non definitiva per autocopy (SQL COPY JOB), che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. L'anteprima pubblica terminerà il 31 luglio 2024. I cluster di anteprima verranno rimossi automaticamente due settimane dopo il termine dell'anteprima. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

Per ulteriori informazioni sull'utilizzo di questo comando, consulta Importazione continua di file da Amazon S3 (anteprima).

Gestisce i comandi COPY per il caricamento dei dati in una tabella. Il comando COPY JOB è un'estensione del comando COPY e automatizza il caricamento dei dati dai bucket Amazon S3. Quando crei un processo COPY, Amazon Redshift rileva quando vengono creati nuovi file Amazon S3 in un percorso specificato e li carica automaticamente senza il tuo intervento. Gli stessi parametri utilizzati nel comando COPY originale vengono utilizzati durante il caricamento dei dati. Amazon Redshift tiene traccia dei file caricati per verificare che vengano caricati una sola volta.

> **Note**
>
> Per informazioni sul comando COPY, inclusi utilizzo, parametri e autorizzazioni, consulta COPY.

Autorizzazione richiesta

Per eseguire il comando COPY di un processo COPY JOB, è necessario disporre del privilegio INSERT della tabella da caricare.

Il ruolo IAM specificato con il comando COPY deve disporre dell'autorizzazione per accedere ai dati da caricare. Per ulteriori informazioni, consulta Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY.

Sintassi

Crea un processo di copia. I parametri del comando COPY vengono salvati con il processo di copia.

```
COPY copy-command JOB CREATE job-name
[AUTO ON | OFF]
```

Modifica la configurazione di un processo di copia.

```
COPY JOB ALTER job-name
[AUTO ON | OFF]
```

Esegui un processo di copia. Vengono utilizzati i parametri del comando COPY memorizzati.

```
COPY JOB RUN job-name
```

Elenca tutti i processi di copia.

```
COPY JOB LIST
```

Mostra i dettagli di un processo di copia.

```
COPY JOB SHOW job-name
```

Elimina un processo di copia.

```
COPY JOB DROP job-name
```

Parametri

copy-command

Un comando COPY che carica i dati da Amazon S3 ad Amazon Redshift. La clausola contiene i parametri COPY che definiscono il bucket Amazon S3, la tabella di destinazione, il ruolo IAM e altri parametri utilizzati durante il caricamento dei dati. Sono supportati tutti i parametri del comando COPY per il caricamento di dati Amazon S3 con le seguenti eccezioni:

- COPY JOB non importa file preesistenti nella cartella a cui fa riferimento il comando COPY. Vengono importati solo i file creati dopo il timestamp di creazione di COPY JOB.
- Non è possibile specificare un comando COPY con le opzioni MAXERROR o IGNOREALLERRORS.
- Non è possibile specificare un file manifesto. COPY JOB richiede una posizione Amazon S3 designata per monitorare i file appena creati.
- Non è possibile specificare un comando COPY con tipi di autorizzazione come le chiavi Access e Secret. Sono supportati solo i comandi COPY che utilizzano il parametro `IAM_ROLE` per l'autorizzazione. Per ulteriori informazioni, consulta Parametri di autorizzazione.
- Il comando COPY JOB non supporta il ruolo IAM predefinito associato al cluster. Devi specificare il `IAM_ROLE` nel comando COPY.

Per ulteriori informazioni, consulta COPY da Amazon S3.

job-name

Nome del processo utilizzato per fare riferimento al processo COPY.

[AUTO ON | OFF]

Clausola che indica se i dati Amazon S3 vengono caricati automaticamente nelle tabelle Amazon Redshift.

- Se è selezionato `ON`, Amazon Redshift monitora il percorso Amazon S3 di origine per individuare i file appena creati e, se ne trova, viene eseguito un comando COPY con i parametri COPY nella definizione del processo. Questa è l'impostazione predefinita.
- Se è selezionato `OFF`, Amazon Redshift non esegue automaticamente il comando COPY JOB.

Note per l'utilizzo

Le opzioni del comando COPY vengono convalidate solo in fase di esecuzione. Ad esempio, un `IAM_ROLE` e un'origine dati Amazon S3 non validi generano errori di runtime all'avvio del processo COPY JOB.

Se il cluster è in pausa, i processi COPY JOB non vengono eseguiti.

Per eseguire query sui file di comando COPY caricati e per gli errori di caricamento, consulta STL_LOAD_COMMITS, STL_LOAD_ERRORS, STL_LOADERROR_DETAIL. Per ulteriori informazioni, consulta Verifica del caricamento corretto dei dati.

Esempi

L'esempio seguente mostra la creazione di un processo COPY JOB per il caricamento di dati da un bucket Amazon S3.

```
COPY public.target_table
FROM 's3://mybucket-bucket/staging-folder'
IAM_ROLE 'arn:aws:iam::123456789012:role/MyLoadRoleName'
JOB CREATE my_copy_job_name
AUTO ON;
```

Documentazione di riferimento dei parametri di COPY

COPY dispone di molti parametri che possono essere utilizzati in molte situazioni. Tuttavia, non tutti i parametri sono supportati in ogni situazione. Ad esempio, per il caricamento da file ORC o PARQUET esiste un numero limitato di parametri supportati. Per ulteriori informazioni, consulta COPY da formati di dati a colonna.

Argomenti

- Origini dati
- Parametri di autorizzazione
- Opzioni di mappatura di colonne
- Parametri del formato dei dati
- Parametri di compressione dei file
- Parametri di conversione dei dati
- Operazioni di caricamento dati
- Elenco alfabetico dei parametri

Origini dati

È possibile caricare i dati da file di testo in un bucket Amazon S3, in un cluster Amazon EMR o in un host remoto a cui il cluster può accedere utilizzando una connessione SSH. È possibile caricare i dati anche direttamente da una tabella DynamoDB.

La dimensione massima di una singola riga di input da qualsiasi origine è pari a 4 MB.

Per esportare i dati da una tabella a un set di file in un Amazon S3, utilizzare il comando UNLOAD.

Argomenti

- COPY da Amazon S3
- COPY da Amazon EMR
- COPY da host remoto (SSH)
- COPY da Amazon DynamoDB

COPY da Amazon S3

Per caricare dati da file ubicati in uno o più bucket S3, utilizzare la clausola FROM per indicare il modo in cui COPY individua i file in Amazon S3. È possibile fornire il percorso dell'oggetto ai file di dati come parte della clausola FROM, oppure è possibile fornire la posizione di un file manifest che contiene un elenco di percorsi dell'oggetto Amazon S3. COPY da Amazon S3 utilizza una connessione HTTPS. Assicurati che gli intervalli IP S3 siano aggiunti all'elenco di indirizzi consentiti. Per ulteriori informazioni sugli intervalli IP S3 richiesti, consulta Isolamento di rete.

> **Important**
>
> Se i bucket Amazon S3 che contengono i file di dati non risiedono nella stessa AWS regione del cluster, devi utilizzare il REGION parametro per specificare la regione in cui si trovano i dati.

Argomenti

- Sintassi
- Esempi
- Parametri facoltativi
- Parametri non supportati

Sintassi

```
FROM { 's3://objectpath' | 's3://manifest_file' }
authorization
| MANIFEST
| ENCRYPTED
| REGION [AS] 'aws-region'
| optional-parameters
```

Esempi

Nell'esempio seguente viene utilizzato un percorso oggetto per caricare dati da Amazon S3.

```
copy customer
from 's3://mybucket/customer'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Nell'esempio seguente viene utilizzato un file manifest per caricare dati da Amazon S3.

```
copy customer
from 's3://mybucket/cust.manifest'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
manifest;
```

Parametri

FROM

L'origine dei dati da caricare. Per ulteriori informazioni sulla codifica del file Amazon S3, consultare Parametri di conversione dei dati.

's3://copy_from_s3_objectpath'

Specifica il percorso degli oggetti Amazon S3 che contengono i dati, ad esempio `'s3://mybucket/custdata.txt'`. Il parametro s3://copy_from_s3_objectpath può fare riferimento a un singolo file o a un insieme di oggetti o cartelle che hanno lo stesso prefisso della chiave. Ad esempio, il nome `custdata.txt` è un prefisso della chiave che si riferisce a un certo numero di file fisici: `custdata.txt`,`custdata.txt.1`, `custdata.txt.2`, `custdata.txt.bak` e così via. Il prefisso della chiave può anche fare riferimento a un certo numero di cartelle. Ad esempio `'s3://mybucket/custfolder'` fa riferimento alle cartelle `custfolder`, `custfolder_1`,

`custfolder_2` e così via. Se un prefisso della chiave fa riferimento a più cartelle, vengono caricati tutti i file in esse contenuti. Se un prefisso della chiave corrisponde sia a un file sia a una cartella, come `custfolder.log`, COPY tenta di caricare anche il file. Se un prefisso della chiave può causare il tentativo da parte di COPY di caricare i file indesiderati, utilizzare un file manifest. Per ulteriori informazioni, consultare copy_from_s3_manifest_file.

> **Important**
>
> Se il bucket S3 che contiene i file di dati non si trova nella stessa AWS regione del cluster, devi utilizzare il REGION parametro per specificare la regione in cui si trovano i dati.

Per ulteriori informazioni, consulta Caricamento di dati da Amazon S3.

's3://copy_from_s3_manifest_file'

Specifica la chiave oggetto Amazon S3 per un file manifest che elenca i file di dati da caricare. L'argomento 's3://copy_from_s3_manifest_file' deve fare esplicito riferimento a un singolo file, ad esempio, `'s3://mybucket/manifest.txt'`. Non può fare riferimento a un prefisso della chiave.

Il manifest è un file di testo in formato JSON che elenca l'URL di ciascun file che deve essere caricato da Amazon S3. L'URL include il nome del bucket e il percorso completo dell'oggetto per il file. I file specificati nel manifesto possono trovarsi in diversi bucket, ma tutti i bucket devono trovarsi nella stessa AWS regione del cluster Amazon Redshift. Se un file viene elencato due volte, viene caricato due volte. L'esempio seguente mostra il JSON per un manifest che carica tre file.

```
{
  "entries": [
    {"url":"s3://mybucket-alpha/custdata.1","mandatory":true},
    {"url":"s3://mybucket-alpha/custdata.2","mandatory":true},
    {"url":"s3://mybucket-beta/custdata.1","mandatory":false}
  ]
}
```

I caratteri delle virgolette doppie sono obbligatori e devono essere virgolette semplici (0x22), non oblique o "intelligenti". Ogni accesso al manifest può facoltativamente includere un flag `mandatory`. Se `mandatory` è impostato come `true`, COPY termina se non trova il file per quella voce; altrimenti COPY continuerà. Il valore predefinito per `mandatory` è `false`.

Quando si carica da file di dati in formato ORC o Parquet, è necessario un campo `meta`, come mostrato nell'esempio seguente.

```
{
   "entries":[
      {
         "url":"s3://mybucket-alpha/orc/2013-10-04-custdata",
         "mandatory":true,
         "meta":{
            "content_length":99
         }
      },
      {
         "url":"s3://mybucket-beta/orc/2013-10-05-custdata",
         "mandatory":true,
         "meta":{
            "content_length":99
         }
      }
   ]
}
```

Il file manifest non deve essere crittografato o compresso, anche se sono specificate le opzioni ENCRYPTED, GZIP, LZOP, BZIP2 o ZSTD. COPY restituisce un errore se il file manifest specificato non viene trovato o non viene formato correttamente.

Se si utilizza un file manifest, il parametro MANIFEST deve essere specificato con il comando COPY. Se il parametro MANIFEST non è specificato, COPY presuppone che il file specificato con FROM sia un file di dati.

Per ulteriori informazioni, consulta Caricamento di dati da Amazon S3.

authorization

Il comando COPY necessita dell'autorizzazione per accedere ai dati in un'altra risorsa AWS , incluso in Amazon S3, Amazon EMR, Amazon DynamoDB e Amazon EC2. Puoi fornire tale autorizzazione facendo riferimento a un ruolo AWS Identity and Access Management (IAM) collegato al cluster (controllo degli accessi basato sui ruoli) o fornendo le credenziali di accesso per un utente (controllo degli accessi basato su chiavi). Per una maggiore sicurezza e flessibilità, consigliamo di utilizzare il controllo degli accessi basato sui ruoli IAM. Per ulteriori informazioni, consulta Parametri di autorizzazione.

MANIFEST

Specifica che un manifest viene utilizzato per identificare i file di dati da caricare da Amazon S3. Se si utilizza il parametro MANIFEST, COPY carica i dati dai file elencati nel manifesto a cui fa riferimento 's3://copy_from_s3_manifest_file'. Se il file manifest non viene trovato o non è formato correttamente, COPY non va a buon fine. Per ulteriori informazioni, consulta Utilizzo di un manifest per specificare i fili di dati.

ENCRYPTED

Una clausola che specifica che i file di input su Amazon S3 sono crittografati utilizzando la crittografia lato client con chiavi simmetriche gestite dal cliente. Per ulteriori informazioni, consulta Caricamento di file di dati crittografati da Amazon S3. Non specificare ENCRYPTED se i file di input sono crittati utilizzando la crittografia lato server di Amazon S3 (SSE-KMS o SSE-S3). COPY legge automaticamente i file crittati lato server.

Se specifichi il parametro ENCRYPTED, devi anche specificare il parametro MASTER_SYMMETRIC_KEY o includere il valore **`master_symmetric_key`** nella stringa CREDENTIALS.

Se i file crittografati sono in formato compresso, aggiungi il parametro GZIP, LZOP, BZIP2 o ZSTD.

I file manifest e JSONPath non devono essere crittografati, anche se è specificata l'opzione ENCRYPTED.

MASTER_SYMMETRIC_KEY 'root_key'

La chiave master simmetrica era utilizzata per crittografare i file di dati su Amazon S3. Se è specificato MASTER_SYMMETRIC_KEY, è necessario specificare anche il parametro ENCRYPTED. MASTER_SYMMETRIC_KEY non può essere utilizzato con il parametro CREDENTIALS. Per ulteriori informazioni, consulta Caricamento di file di dati crittografati da Amazon S3.

Se i file crittografati sono in formato compresso, aggiungi il parametro GZIP, LZOP, BZIP2 o ZSTD.

REGION [AS] 'aws-region'

Speciifica la AWS regione in cui si trovano i dati di origine. REGION è obbligatorio per COPY da un bucket Amazon S3 o una tabella DynamoDB quando la risorsa AWS che contiene i dati non si trova nella stessa regione del cluster Amazon Redshift.

Il valore per aws_region deve corrispondere a una regione elencata nella tabella Regioni ed endpoint di Amazon Redshift.

Se viene specificato il parametro REGION, tutte le risorse, compresi un file manifest o più bucket Amazon S3 devono trovarsi nella regione specificata.

> **Note**
>
> Il trasferimento di dati tra regioni comporta costi aggiuntivi a carico del bucket Amazon S3 o della tabella DynamoDB che contiene i dati. Per ulteriori informazioni sui prezzi, consulta Data Transfer OUT da Amazon S3 a un'altra AWS regione nella pagina dei prezzi di Amazon S3 e Data Transfer OUT nella pagina dei prezzi di Amazon DynamoDB.

Per impostazione predefinita, COPY presuppone che i dati si trovino nella stessa regione del cluster Amazon Redshift.

Parametri facoltativi

Facoltativamente è possibile specificare i seguenti parametri con COPY da Amazon S3:

- Opzioni di mappatura di colonne
- Parametri del formato dei dati
- Parametri di conversione dei dati
- Operazioni di caricamento dati

Parametri non supportati

Non è possibile utilizzare i seguenti parametri con COPY da Amazon S3:

- SSH
- READRATIO

COPY da Amazon EMR

È possibile utilizzare il comando COPY per caricare dati in parallelo da un cluster Amazon EMR configurato per scrivere file di testo nel Hadoop Distributed File System (HDFS) del cluster sotto forma di file a larghezza fissa, delimitati da caratteri, CSV, formattati JSON o Avro.

Argomenti

Sintassi

```
FROM 'emr://emr_cluster_id/hdfs_filepath'
authorization
[ optional_parameters ]
```

Esempio

Nell'esempio seguente i dati vengono caricati da un cluster Amazon EMR.

```
copy sales
from 'emr://j-SAMPLE2B500FC/myoutput/part-*'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Parametri

FROM

L'origine dei dati da caricare.

'emr://emr_cluster_id/hdfs_file_path'

L'identificatore univoco per il cluster Amazon EMR e il percorso del file HDFS che fa riferimento ai file di dati per il comando COPY. I nomi dei file di dati HDFS non devono contenere i caratteri jolly asterisco (*) e punto interrogativo (?).

> **Note**
>
> Il cluster Amazon EMR deve continuare a funzionare fino al completamento dell'operazione COPY. Se uno qualsiasi dei file di dati HDFS viene modificato o cancellato prima del completamento dell'operazione COPY, si potrebbero ottenere risultati imprevisti o l'operazione COPY potrebbe fallire.

È possibile utilizzare i caratteri jolly asterisco (*) e punto interrogativo (?) come parte dell'argomento hdfs_file_path del nome file. Ad esempio `'emr://j-SAMPLE2B500FC/myoutput/part*'` identifica i file `part-0000`, `part-0001` e così via. Se il percorso del file non contiene caratteri jolly, viene trattato come una stringa letterale. Se specifichi solo il nome di una cartella, COPY tenta di caricare tutti i file nella cartella.

> **Important**
>
> Se utilizzi caratteri jolly o solo il nome della cartella, verifica che non vengano caricati file indesiderati. Ad esempio, alcuni processi potrebbero scrivere un file di log nella cartella di output.

Per ulteriori informazioni, consulta Caricamento di dati da Amazon EMR.

authorization

Il comando COPY necessita dell'autorizzazione per accedere ai dati in un'altra risorsa AWS , incluso in Amazon S3, Amazon EMR, Amazon DynamoDB e Amazon EC2. Puoi fornire tale autorizzazione facendo riferimento a un ruolo AWS Identity and Access Management (IAM) collegato al cluster (controllo degli accessi basato sui ruoli) o fornendo le credenziali di accesso per un utente (controllo degli accessi basato su chiavi). Per una maggiore sicurezza e flessibilità, consigliamo di utilizzare il controllo degli accessi basato sui ruoli IAM. Per ulteriori informazioni, consulta Parametri di autorizzazione.

Parametri supportati

Facoltativamente è possibile specificare i seguenti parametri con COPY da Amazon EMR:

- Opzioni di mappatura di colonne
- Parametri del formato dei dati
- Parametri di conversione dei dati
- Operazioni di caricamento dati

Parametri non supportati

Non è possibile utilizzare i seguenti parametri con COPY da Amazon EMR:

- ENCRYPTED

- MANIFEST
- REGION
- READRATIO
- SSH

COPY da host remoto (SSH)

È possibile utilizzare il comando COPY per caricare dati in parallelo da uno o più host remoti, quali istanze Amazon Elastic Compute Cloud (Amazon EC2) o altri computer. COPY si connette agli host remoti utilizzando Secure Shell (SSH) ed esegue comandi sugli host remoti per generare output di testo. L'host remoto può essere un'istanza EC2 Linux o un altro computer Unix o Linux configurato per accettare connessioni SSH. Amazon Redshift può connettersi a più host e può aprire più connessioni SSH per ogni host. Amazon Redshift invia un comando univoco attraverso ogni connessione per generare output di testo per l'output standard dell'host, che legge quindi come un file di testo.

Utilizzare la clausola FROM per specificare la chiave oggetto di Amazon S3 per il file manifest che fornisce le informazioni che COPY utilizza per aprire le connessioni SSH ed eseguire i comandi remoti.

Argomenti

- Sintassi
- Esempi
- Parametri
- Parametri facoltativi
- Parametri non supportati

> **Important**
>
> Se il bucket S3 che contiene i file manifest non si trova nella stessa regione AWS del cluster, è necessario utilizzare il parametro REGION per specificare la regione in cui si trova il bucket.

Sintassi

```
FROM 's3://'ssh_manifest_file' }
```

```
authorization
SSH
| optional-parameters
```

Esempi

Nell'esempio seguente viene utilizzato un file manifest per caricare dati da un host remoto tramite SSH.

```
copy sales
from 's3://mybucket/ssh_manifest'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
ssh;
```

Parametri

FROM

L'origine dei dati da caricare.

's3://copy_from_ssh_manifest_file'

Il comando COPY può connettersi a più host tramite SSH e può creare più connessioni SSH per ogni host. COPY esegue un comando attraverso ogni connessione host, quindi carica l'output dai comandi in parallelo nella tabella. L'argomento s3://copy_from_ssh_manifest_file specifica la chiave oggetto di Amazon S3 per il file manifest che fornisce le informazioni che COPY utilizza per aprire connessioni SSH ed eseguire i comandi remoti.

L'argomento s3://copy_from_ssh_manifest_file deve fare esplicito riferimento a un singolo file; non può essere un prefisso della chiave. Di seguito viene riportato un esempio:

```
's3://mybucket/ssh_manifest.txt'
```

Il file manifest è un file di testo in formato JSON che Amazon Redshift utilizza per la connessione all'host. Il file manifesto specifica gli endpoint dell'host SSH e i comandi che verranno eseguiti sugli host per restituire i dati ad Amazon Redshift. Facoltativamente, puoi includere la chiave pubblica dell'host, il nome utente di login e un flag obbligatorio per ogni voce. L'esempio seguente mostra un file manifest che crea due connessioni SSH:

```
{
```

```
    "entries": [
     {"endpoint":"<ssh_endpoint_or_IP>",
           "command": "<remote_command>",
           "mandatory":true,
           "publickey": "<public_key>",
           "username": "<host_user_name>"},
     {"endpoint":"<ssh_endpoint_or_IP>",
           "command": "<remote_command>",
           "mandatory":true,
           "publickey": "<public_key>",
           "username": "<host_user_name>"}
     ]
}
```

Il file manifest contiene un costrutto `"entries"` per ogni connessione SSH. È possibile avere più connessioni a un singolo host o più connessioni a più host. I caratteri delle doppie virgolette sono richiesti come mostrato, sia per i nomi dei campi sia per i valori. I caratteri delle virgolette devono essere virgolette semplici (0x22), non oblique o "intelligenti". L'unico valore che non necessita di caratteri di virgolette doppie è il valore booleano `true` o `false` per il campo `"mandatory"`.

Nella seguente lista sono descritti i campi del file manifest.

endpoint

L'indirizzo URL o l'indirizzo IP dell'host, ad esempio `"ec2-111-222-333.compute-1.amazonaws.com"` o `"198.51.100.0"`.

command

Il comando che deve essere eseguito dall'host per generare output di testo o binario in formato gzip, lzop, bzip2 o zstd. Il comando può essere qualsiasi comando che l'utente "host_user_name" è autorizzato a eseguire. Il comando può essere semplice come stampare un file o eseguire una query su un database o lanciare uno script. L'output (file di testo o file binario gzip, lzop o bzip2) deve essere in un formato che il comando COPY di Amazon Redshift possa importare. Per ulteriori informazioni, consulta Preparazione dei dati di input.

publickey

(Facoltativo) La chiave pubblica dell'host. Se fornita, Amazon Redshift userà la chiave pubblica per identificare l'host. Se la chiave pubblica non viene fornita, Amazon Redshift non proverà a eseguire l'identificazione dell'host. Ad esempio, se la chiave pubblica dell'host remoto è `ssh-rsa AbcCbaxxx…Example root@amazon.com`, digita il seguente testo nel campo della chiave pubblica: `"AbcCbaxxx…Example"`

mandatory

(Facoltativo) Una clausola che indica se il comando COPY debba fallire se il tentativo di connessione fallisce. Il valore predefinito è `false`. Se Amazon Redshift non riesce a effettuare almeno una connessione, il comando COPY avrà esito negativo.

username

(Facoltativo) Il nome utente che verrà utilizzato per accedere al sistema host ed eseguire il comando remoto. Il nome di accesso dell'utente deve essere lo stesso usato per aggiungere la chiave pubblica del cluster Amazon Redshift al file delle chiavi autorizzate dell'host. Il nome utente predefinito è `redshift`.

Per ulteriori informazioni sulla creazione di un file manifest, consultare Processo di caricamento dei dati.

Per utilizzare COPY da un host remoto, il parametro SSH deve essere specificato con il comando COPY. Se il parametro SSH non è specificato, COPY presuppone che il file specificato con FROM sia un file di dati e non va a buon fine.

Se si utilizza la compressione automatica, il comando COPY esegue due operazioni di lettura dati, il che significa che eseguirà il comando remoto due volte. La prima operazione di lettura consiste nel fornire un campione di dati per l'analisi della compressione e la seconda operazione di lettura carica effettivamente i dati. Se la doppia esecuzione del comando remoto potrebbe causare un problema, è necessario disattivare la compressione automatica. Per disattivare la compressione automatica, esegui il comando COPY con il parametro COMPUPDATE impostato su OFF. Per ulteriori informazioni, consulta Caricamento di tabelle con compressione automatica.

Per le procedure dettagliate sull'utilizzo di COPY da SSH, consultare Caricamento di dati da host remoti.

authorization

Il comando COPY necessita dell'autorizzazione per accedere ai dati in un'altra risorsa AWS , incluso in Amazon S3, Amazon EMR, Amazon DynamoDB e Amazon EC2. È possibile fornire tale autorizzazione facendo riferimento a un ruolo AWS Identity and Access Management (IAM) collegato al cluster (controllo degli accessi basato sui ruoli) o fornendo le credenziali di accesso per un utente (controllo degli accessi basato su chiavi). Per una maggiore sicurezza e flessibilità, consigliamo di utilizzare il controllo degli accessi basato sui ruoli IAM. Per ulteriori informazioni, consulta Parametri di autorizzazione.

SSH

Una clausola che specifica che i dati devono essere caricati da un host remoto utilizzando il protocollo SSH. Se specifichi SSH. devi anche fornire un file manifest usando l'argomento s3://copy_from_ssh_manifest_file.

> **Note**
>
> Se stai utilizzando SSH per effettuare la copia da un host utilizzando un indirizzo IP privato su un VPC remoto, il VPC deve avere il routing VPC avanzato abilitato. Per ulteriori informazioni sul routing VPC avanzato, consultare Amazon Redshift Spectrum con il routing VPC avanzato.

Parametri facoltativi

Facoltativamente è possibile specificare i seguenti parametri con COPY da SSH:

- Opzioni di mappatura di colonne
- Parametri del formato dei dati
- Parametri di conversione dei dati
- Operazioni di caricamento dati

Parametri non supportati

Non è possibile utilizzare i seguenti parametri con COPY da SSH:

- ENCRYPTED
- MANIFEST
- READRATIO

COPY da Amazon DynamoDB

Per caricare i dati da una tabella DynamoDB esistente, utilizza la clausola FROM per specificare il nome della tabella DynamoDB.

Argomenti

- Sintassi

- Esempi
- Parametri facoltativi
- Parametri non supportati

> **Important**
>
> Se la tabella DynamoDB non si trova nella stessa regione del cluster Amazon Redshift è necessario utilizzare il parametro REGION per specificare la regione in cui si trovano i dati.

Sintassi

```
FROM 'dynamodb://table-name'
authorization
READRATIO ratio
| REGION [AS] 'aws_region'
| optional-parameters
```

Esempi

Nell'esempio seguente i dati sono caricati da una tabella DynamoDB.

```
copy favoritemovies from 'dynamodb://ProductCatalog'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
readratio 50;
```

Parametri

FROM

L'origine dei dati da caricare.

'dynamodb://table-name'

Il nome della tabella DynamoDB che contiene i dati, ad esempio `'dynamodb://ProductCatalog'`. Per maggiori dettagli su come gli attributi DynamoDB siano mappati alle colonne Amazon Redshift, consultare Caricamento di dati da una tabella Amazon DynamoDB.

Il nome di una tabella DynamoDB è univoco per AWS un account, identificato dalle AWS credenziali di accesso.

authorization

Il comando COPY necessita dell'autorizzazione per accedere ai dati in un'altra risorsa AWS , incluso in Amazon S3, Amazon EMR, Amazon DynamoDB e Amazon EC2. È possibile fornire tale autorizzazione facendo riferimento a un ruolo AWS Identity and Access Management (IAM) collegato al cluster (controllo degli accessi basato sui ruoli) o fornendo le credenziali di accesso per un utente (controllo degli accessi basato su chiavi). Per una maggiore sicurezza e flessibilità, consigliamo di utilizzare il controllo degli accessi basato sui ruoli IAM. Per ulteriori informazioni, consulta Parametri di autorizzazione.

READRATIO [AS] ratio

La percentuale di throughput assegnato della tabella DynamoDB da utilizzare per il caricamento dei dati. READRATIO è necessario per COPY da DynamoDB. Non può essere utilizzato con COPY da Amazon S3. È consigliabile impostare la percentuale su un valore minore del throughput assegnato mediamente non utilizzato. I valori validi sono numeri interi compresi tra 1 e 200.

> **Important**
>
> Impostando READRATIO su 100 o su un valore superiore si consente a Amazon Redshift di consumare l'intero throughput assegnato della tabella DynamoDB che degrada notevolmente le prestazioni delle operazioni di lettura simultanee rispetto alla stessa tabella durante la sessione di COPY. Il traffico della funzione di scrittura non viene modificato. I valori superiori a 100 possono risolvere problemi in scenari rari quando Amazon Redshift non è in grado di soddisfare il throughput assegnato della tabella. Se i dati vengono caricati da DynamoDB ad Amazon Redshift su base continuativa, considerare l'organizzazione delle tabelle DynamoDB come una serie temporale per separare il traffico in tempo reale dall'operazione COPY.

Parametri facoltativi

Facoltativamente è possibile specificare i seguenti parametri con COPY da Amazon DynamoDB:

- Opzioni di mappatura di colonne
- Sono supportati i seguenti parametri di conversione dei dati:
  - ACCEPTANYDATE
  - BLANKSASNULL

  - DATEFORMAT
  - EMPTYASNULL
  - ROUNDEC
  - TIMEFORMAT
  - TRIMBLANKS
  - TRUNCATECOLUMNS
- Operazioni di caricamento dati

Parametri non supportati

Non è possibile utilizzare i seguenti parametri con COPY da DynamoDB:

- Tutti i parametri del formato dei dati
- ESCAPE
- FILLRECORD
- IGNOREBLANKLINES
- IGNOREHEADER
- NULL
- REMOVEQUOTES
- ACCEPTINVCHARS
- MANIFEST
- ENCRYPTED

Parametri di autorizzazione

Il comando COPY richiede l'autorizzazione per accedere ai dati in un'altra AWS risorsa, tra cui Amazon S3, Amazon EMR, Amazon DynamoDB e Amazon EC2. Per concedere questa autorizzazione, è necessario fare riferimento a un ruolo (IAM)AWS Identity and Access Management associato al cluster (controllo degli accessi basato sul ruolo). Puoi crittografare i dati di caricamento su Amazon S3.

I seguenti argomenti forniscono ulteriori dettagli ed esempi di opzioni di autenticazione:

- Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY

- Controllo degli accessi basato sui ruoli
- Controllo degli accessi basato su chiave

Utilizza una delle seguenti opzioni per fornire l'autorizzazione per il comando COPY:

- Parametro IAM_ROLE
- Parametri ACCESS_KEY_ID and SECRET_ACCESS_KEY
- CREDENTIALSClausola

IAM_ROLE {default | 'arn:aws:iam::*<Account AWS-id>*:ruolo/*<role-name>*' }

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando COPY.

L'Amazon Resource Name (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione. Se specifichi IAM_ROLE, non è possibile utilizzare ACCESS_KEY_ID e SECRET_ACCESS_KEY, SESSION_TOKEN o CREDENTIALS.

Di seguito è mostrata la sintassi del parametro IAM_ROLE.

```
IAM_ROLE { default | 'arn:aws:iam::<Account AWS-id>:role/<role-name>' }
```

Per ulteriori informazioni, consulta Controllo degli accessi basato sui ruoli.

ACCESS_KEY_ID '' SECRET_ACCESS_KEY 'chiave di accesso segreta' access-key-id

Questo metodo di autorizzazione non è raccomandato.

> **Note**
>
> Invece di fornire le credenziali di accesso come testo in chiaro, consigliamo vivamente di utilizzare l'autenticazione basata sui ruoli specificando il parametro IAM_ROLE. Per ulteriori informazioni, consulta Controllo degli accessi basato sui ruoli.

SESSION_TOKEN 'temporary-token'

Il token di sessione da utilizzare con le credenziali di accesso temporaneo. Quando è specificato SESSION_TOKEN, devi utilizzare anche ACCESS_KEY_ID e SECRET_ACCESS_KEY per fornire

credenziali di chiave di accesso temporanee. Se specifichi SESSION_TOKEN, non è possibile utilizzare IAM_ROLE o CREDENTIALS. Per ulteriori informazioni, consultare Credenziali di sicurezza temporanee nella Guida per l'utente di IAM.

> **Note**
>
> Invece di creare credenziali di sicurezza temporanee, consigliamo vivamente di utilizzare l'autenticazione basata su ruoli. Quando si autorizza l'uso di un ruolo IAM, Amazon Redshift crea automaticamente credenziali utente temporanee per ogni sessione. Per ulteriori informazioni, consulta Controllo degli accessi basato sui ruoli.

Di seguito è riportata la sintassi del parametro SESSION_TOKEN con i parametri ACCESS_KEY_ID e SECRET_ACCESS_KEY.

```
ACCESS_KEY_ID '<access-key-id>'
SECRET_ACCESS_KEY '<secret-access-key>'
SESSION_TOKEN '<temporary-token>';
```

Se specifichi SESSION_TOKEN, non è possibile utilizzare CREDENTIALS o IAM_ROLE.

[WITH] CREDENTIALS [AS] 'credentials-args'

Una clausola che indica il metodo che il cluster utilizzerà per accedere ad altre risorse che contengono file di dati o file manifest. AWS Non è possibile utilizzare il parametro CREDENTIALS con IAM_ROLE o ACCESS_KEY_ID e SECRET_ACCESS_KEY.

> **Note**
>
> Per una maggiore flessibilità, consigliamo di utilizzare il parametro IAM_ROLE invece del parametro CREDENTIALS.

Facoltativamente, se utilizzi il parametro ENCRYPTED, la stringa credentials-args fornisce anche la chiave di crittografia.

La stringa credentials-args prevede la distinzione tra maiuscole e minuscole e non deve contenere spazi.

Le parole chiave WITH e AS sono opzionali e vengono ignorate.

È possibile specificare role-based access control o key-based access control. In entrambi i casi, l'utente o il ruolo IAM deve disporre delle autorizzazioni necessarie per accedere alle risorse AWS specificate. Per ulteriori informazioni, consulta Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY.

> **Note**
>
> Per salvaguardare le AWS credenziali e proteggere i dati sensibili, consigliamo vivamente di utilizzare il controllo degli accessi basato sui ruoli.

Per specificare il controllo degli accessi basato su ruoli, fornisci la stringa credentials-args nel seguente formato.

```
'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
```

Per utilizzare le credenziali del token temporanee, devi fornire l'ID chiave di accesso, la chiave di accesso segreta e il token temporanei. La stringa credentials-args è nel seguente formato.

```
CREDENTIALS
'aws_access_key_id=<temporary-access-key-id>;aws_secret_access_key=<temporary-
secret-access-key>;token=<temporary-token>'
```

Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee.

Se utilizzi il parametro ENCRYPTED, la stringa credentials-args è nel seguente formato, dove *<master-key>* è il valore della chiave master che è stata usata per crittare i file.

```
CREDENTIALS
'<credentials-args>;master_symmetric_key=<root-key>'
```

Ad esempio, il seguente comando COPY utilizza un controllo di accesso basato su ruoli con una chiave di crittografia.

```
copy customer from 's3://mybucket/mydata'
credentials
'aws_iam_role=arn:aws:iam::<account-id>:role/<role-name>;master_symmetric_key=<root-
key>'
```

Il seguente comando COPY mostra il controllo di accesso basato sul ruolo con una chiave di crittografia.

```
copy customer from 's3://mybucket/mydata'
credentials
'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-
name>;master_symmetric_key=<root-key>'
```

Opzioni di mappatura di colonne

Per impostazione predefinita, COPY inserisce i valori nelle colonne della tabella di destinazione nello stesso ordine in cui i campi vengono visualizzati nei file di dati. Se l'ordine predefinito delle colonne non funziona, è possibile specificare un elenco di colonne o utilizzare delle espressioni JSONPath per mappare i campi dei dati di origine nelle colonne di destinazione.

- Column List
- JSONPaths File

Elenco di colonne

È possibile specificare un elenco separato da virgole di nomi di colonna per caricare i campi dati sorgente in specifiche colonne di destinazione. Le colonne possono essere in qualsiasi ordine nell'istruzione COPY, ma quando si caricano da flat file, come in un bucket Amazon S3, il loro ordine deve corrispondere all'ordine dei dati di origine.

Durante il caricamento da una tabella Amazon DynamoDB l'ordine non è importante. Il comando COPY fa corrispondere i nomi degli attributi negli elementi recuperati dalla tabella DynamoDB ai nomi di colonna nella tabella Amazon Redshift. Per ulteriori informazioni, consultare Caricamento di dati da una tabella Amazon DynamoDB

Il formato dell'elenco di colonne è il seguente.

```
COPY tablename (column1 [,column2, ...])
```

Se una colonna nella tabella di destinazione viene omessa dall'elenco delle colonne, COPY carica l'espressione DEFAULT della colonna di destinazione.

Se la colonna di destinazione non ha un valore predefinito, COPY tenta di caricare NULL.

Se COPY tenta di assegnare NULL a una colonna definita come NOT NULL, il comando COPY non viene eseguito.

Se una colonna IDENTITY è inclusa nell'elenco di colonne, allora è necessario specificare anche EXPLICIT_IDS; se viene omessa una colonna IDENTITY, allora EXPLICIT_IDS non può essere specificato. Se non è specificato un elenco di colonne, il comando si comporta come se fosse specificato un elenco completo di colonne in ordine, con le colonne IDENTITY omesse se non era specificato anche EXPLICIT_IDS.

Se una colonna è definita con GENERATED BY DEFAULT AS IDENTITY, può esser copiata. i valori vengono generati o aggiornati con i valori forniti dall'utente. L'opzione EXPLICIT_IDS non è obbligatoria. COPY non aggiorna la filigrana ad elevata identità. Per ulteriori informazioni, consulta GENERATED BY DEFAULT AS IDENTITY.

File JSONPath

Quando si carica da file di dati in formato JSON o Avro, COPY mappa automaticamente gli elementi dei dati nei dati di origine JSON o Avro alle colonne della tabella di destinazione. Lo fa associando i nomi dei campi nello schema Avro ai nomi delle colonne nella tabella di destinazione o nell'elenco delle colonne.

In alcuni casi, i nomi delle colonne e dei campi non corrispondono o è necessario mappare a livelli più profondi nella gerarchia dei dati. In questi casi, per mappare esplicitamente gli elementi di dati JSON o Avro alle colonne, è possibile utilizzare un file JSONPaths.

Per ulteriori informazioni, consulta File JSONPath.

Parametri del formato dei dati

Per impostazione predefinita, il comando COPY prevede che i dati di origine siano testi UTF-8 delimitati da caratteri. Il delimitatore predefinito è una barra verticale (|). Se i dati di origine sono in un altro formato, utilizza i seguenti parametri per specificare il formato dei dati:

- FORMAT
- CSV
- DELIMITER
- FIXEDWIDTH
- SHAPEFILE

- AVRO
- JSON
- PARQUET
- ORC

Oltre ai formati di dati standard, COPY supporta i seguenti formati di dati a colonna per COPY da Amazon S3:

- ORC
- PARQUET

COPY dal formato a colonne è supportato con alcune restrizioni. Per ulteriori informazioni, consulta COPY da formati di dati a colonna.

Parametri del formato dei dati

FORMAT [AS]

(Facoltativo) Identifica le parole chiave del formato dei dati. Gli argomenti FORMAT sono descritti di seguito.

CSV [QUOTE [AS] 'quote_character']

Consente l'utilizzo del formato CSV nei dati di input. Per creare automaticamente sequenza di escape con i delimitatori, i caratteri di linea e i ritorni a capo, racchiudi il campo nel carattere specificato dal parametro QUOTE. Il carattere virgoletta di default è una virgoletta doppia ("). Quando si utilizza il carattere virgolette in un campo, eseguire l'escape del carattere aggiungendo altre virgolette. Ad esempio, se il carattere virgoletta è una virgoletta doppia, per inserire la stringa `A "quoted" word` il file di input deve includere la stringa `"A ""quoted"" word"`. Quando si utilizza il parametro CSV, il delimitatore predefinito è una virgola (,). È possibile specificare un delimitatore diverso utilizzando il parametro DELIMITER.

Quando un campo è racchiuso tra virgolette, lo spazio tra i delimitatori e i caratteri delle virgolette viene ignorato. Se il delimitatore è un carattere di spazio, ad esempio un carattere di tabulazione, il delimitatore non viene trattato come spazio.

Il CSV non può essere utilizzato con FIXEDWIDTH, REMOVEQUOTES o ESCAPE.

QUOTE [AS] 'quote_character'

Facoltativo. Specifica il carattere da utilizzare come carattere virgoletta quando si utilizza il parametro CSV. Il carattere predefinito è una doppia virgoletta ("). Se si utilizzi il parametro QUOTE per definire un carattere virgoletta diverso dalla doppia virgoletta, non è necessario creare una sequenza di escape con le virgolette doppie all'interno del campo. Il parametro QUOTE può essere utilizzato solo con il parametro CSV. La parola chiave AS è facoltativa.

DELIMITER [AS] ['delimiter_char']

Specifica il singolo carattere ASCII utilizzato per separare i campi del file di input, ad esempio un carattere pipe (|), una virgola (,) o una tabulazione (\t). Sono supportati caratteri ASCII non stampati. I caratteri ASCII possono anche essere rappresentati in ottale, usando il formato '\ddd', dove 'd' è una cifra ottale (0-7). Il delimitatore predefinito è un carattere pipe (|), a meno che non venga utilizzato il parametro CSV, nel qual caso il delimitatore predefinito è una virgola (,). La parola chiave AS è facoltativa. DELIMITER non può essere utilizzato con FIXEDWIDTH.

FIXEDWIDTH 'fixedwidth_spec'

Carica i dati da un file in cui ogni larghezza di colonna è fissa, invece di avere colonne separate da un delimitatore. La stringa fixedwidth_spec specifica l'etichetta e la larghezza della colonna definite dall'utente. L'etichetta della colonna può essere sia una stringa di testo sia un intero, a seconda di ciò che sceglie l'utente. L'etichetta della colonna non ha alcuna relazione con il nome della colonna. L'ordine delle coppie etichetta/larghezza deve corrispondere esattamente all'ordine delle colonne della tabella. FIXEDWIDTH non può essere utilizzato con CSV o DELIMITER. In Amazon Redshift, la lunghezza delle colonne CHAR e VARCHAR è espressa in byte, quindi assicurati che la larghezza della colonna specificata corrisponda alla lunghezza binaria dei caratteri multibyte durante la preparazione del file da caricare. Per ulteriori informazioni, consulta Tipi di carattere.

Di seguito viene mostrato il formato per fixedwidth_spec:

```
'colLabel1:colWidth1,colLabel:colWidth2, ...'
```

SHAPEFILE [SIMPLIFY [AUTO] ['tolleranza']]

Consente l'utilizzo del formato SHAPEFILE nei dati di input. Per impostazione predefinita, la prima colonna dello shapefile è una colonna `GEOMETRY` o `IDENTITY`. Tutte le colonne successive seguono l'ordine specificato nello shapefile.

Non è possibile utilizzare SHAPEFILE con FIXEDWIDTH, REMOVEQUOTES o ESCAPE.

Per utilizzare oggetti GEOGRAPHY con COPY FROM SHAPEFILE, prima importa in un colonna GEOMETRY, quindi converti gli oggetti in oggetti GEOGRAPHY.

SIMPLIFY [tolleranza]

(Facoltativo) Semplifica tutte le geometrie durante il processo di importazione dati utilizzando l'algoritmo Ramer-Douglas-Peucker e la tolleranza specificata.

SIMPLIFY AUTO [tolleranza]

(Facoltativo) Semplifica solo le geometrie più grandi della dimensione massima della geometria. Questa semplificazione utilizza l'algoritmo Ramer-Douglas-Peucker e la tolleranza calcolata automaticamente se questa non supera la tolleranza specificata. Questo algoritmo calcola la dimensione per memorizzare gli oggetti entro la tolleranza specificata. Il valore tolleranza è facoltativo.

Per esempi di caricamento di uno shapefile, consultare Caricamento di uno shapefile in Amazon Redshift.

AVRO [AS] 'avro_option'

Specifica che i dati sorgente sono in formato Avro.

Il formato Avro è supportato per COPY da questi servizi e protocolli:

- Amazon S3
- Amazon EMR
- Host remoto (SSH)

Avro non è supportato per COPY da DynamoDB.

Avro è un protocollo di serializzazione dei dati. Un file sorgente Avro include uno schema che definisce la struttura dei dati. Il tipo di schema Avro deve essere `record`. COPY accetta file Avro creati usando il codec di default non compresso così come i codec di compressione `deflate` e `snappy`. Per ulteriori informazioni su Avro, consultare Apache Avro.

I valori validi per avro_option sono i seguenti:

- `'auto'`
- `'auto ignorecase'`
- `'s3://jsonpaths_file'`

Il valore predefinito è `'auto'`.

COPY mappa automaticamente gli elementi di dati nei dati di origine Avro alle colonne della tabella di destinazione. Lo fa associando i nomi dei campi nello schema Avro ai nomi delle colonne nella tabella di destinazione. La corrispondenza prevede una distinzione tra lettere maiuscole e minuscole per `'auto'` e non fa distinzione tra lettere maiuscole e minuscole per `'auto ignorecase'`.

I nomi delle colonne nelle tabelle Amazon Redshift sono sempre in minuscolo, quindi quando si utilizza l'opzione `'auto'`, anche i nomi dei campi corrispondenti devono essere in minuscolo. Se i nomi dei campi non sono tutti minuscoli, è possibile utilizzare l'opzione `'auto ignorecase'`. Con l'argomento `'auto'` di default, COPY riconosce solo il primo livello di campi, o campi esterni, nella struttura.

Per mappare esplicitamente i nomi delle colonne ai nomi dei campi Avro, è possibile utilizzare un File JSONPath.

Per impostazione predefinita, COPY tenta di far corrispondere tutte le colonne della tabella di destinazione ai nomi dei campi Avro. Per caricare un sottoinsieme delle colonne, è possibile specificare facoltativamente un elenco di colonne. Se una colonna nella tabella di destinazione viene omessa dall'elenco delle colonne, COPY carica l'espressione DEFAULT della colonna di destinazione. Se la colonna di destinazione non ha un valore di default, COPY prova a caricare NULL. Se una colonna è inclusa nell'elenco di colonne e COPY non trova un campo corrispondente nei dati Avro, COPY prova a caricare NULL nella colonna.

Se COPY tenta di assegnare NULL a una colonna definita come NOT NULL, il comando COPY non viene eseguito.

Schema Avro

Un file di dati sorgente Avro include uno schema che definisce la struttura dei dati. COPY legge lo schema che fa parte del file di dati sorgente Avro per mappare gli elementi di dati sulle colonne della tabella di destinazione. L'esempio seguente mostra uno schema Avro.

```
{
    "name": "person",
    "type": "record",
    "fields": [
        {"name": "id", "type": "int"},
        {"name": "guid", "type": "string"},
        {"name": "name", "type": "string"},
        {"name": "address", "type": "string"}]
```

```
}
```

Lo schema Avro è definito utilizzando il formato JSON. L'oggetto JSON di livello superiore contiene tre coppie nome-valore con i nomi o le chiavi, `"name"`, `"type"` e `"fields"`.

Le chiavi `"fields"` si accoppiano con un array di oggetti che definiscono il nome e il tipo di dati di ogni campo nella struttura dei dati. Per impostazione predefinita, COPY fa corrispondere automaticamente i nomi dei campi ai nomi delle colonne. I nomi delle colonne sono sempre in minuscolo, quindi anche i nomi dei campi corrispondenti devono essere in minuscolo a meno che non si specifichi l'opzione `'auto ignorecase'`. I nomi dei campi che non corrispondono al nome di una colonna vengono ignorati. L'ordine non ha importanza. Nell'esempio precedente COPY mappa i nomi delle colonne `id`, `guid`, `name` e `address`.

Con l'argomento predefinito `'auto'`, COPY fa corrispondere solo agli oggetti di primo livello alle colonne. Per mappare a livelli più profondi nello schema o se i nomi dei campi e delle colonne non corrispondono, utilizza un file JSONPath per definire la mappatura. Per ulteriori informazioni, consulta File JSONPath.

Se il valore associato a una chiave è un tipo di dati Avro complesso come byte, array, record, mappa o link, COPY carica il valore come stringa. Qui, la stringa è la rappresentazione JSON dei dati. COPY carica i tipi di dati enum Avro come stringhe, dove il contenuto è il nome del tipo. Per un esempio, consultare COPY dal formato JSON.

La dimensione massima dell'intestazione del file Avro, che include lo schema e i metadati del file, è di 1 MB.

La dimensione massima di un singolo blocco dati Avro è di 4 MB. È diverso dalla dimensione massima della riga. Se si supera la dimensione massima di un singolo blocco dati Avro, anche se la dimensione della riga risultante è inferiore al limite di 4 MB, il comando COPY non viene eseguito.

Nel calcolo delle dimensioni delle righe, Amazon Redshift internamente conta due volte i caratteri pipe (|). Se i dati immessi contengono un numero molto elevato di caratteri pipe, è possibile che la dimensione delle righe superi i 4 MB anche se il blocco di dati è inferiore a 4 MB.

JSON [AS] 'json_option'

I dati sorgente sono in formato JSON.

Il formato JSON è supportato per COPY da questi servizi e protocolli:

- Amazon S3

- COPY da Amazon EMR
- COPY da SSH

JSON non è supportato per COPY da DynamoDB.

I valori validi per json_option sono i seguenti:

- `'auto'`
- `'auto ignorecase'`
- `'s3://jsonpaths_file'`
- `'noshred'`

Il valore predefinito è `'auto'`. Amazon Redshift non divide gli attributi delle strutture JSON in più colonne durante il caricamento di un documento JSON.

Per impostazione predefinita, COPY tenta di far corrispondere tutte le colonne della tabella di destinazione alle chiavi dei nomi dei campi JSON. Per caricare un sottoinsieme delle colonne, è possibile specificare facoltativamente un elenco di colonne. Se le chiavi del nome del campo JSON non sono tutte in minuscolo, è possibile utilizzare l'opzione `'auto ignorecase'` o un File JSONPath per mappare esplicitamente i nomi delle colonne sulle chiavi dei nomi dei campi JSON.

Se una colonna nella tabella di destinazione viene omessa dall'elenco delle colonne, COPY carica l'espressione DEFAULT della colonna di destinazione. Se la colonna di destinazione non ha un valore di default, COPY prova a caricare NULL. Se una colonna è inclusa nell'elenco di colonne e COPY non trova un campo corrispondente nei dati JSON, COPY prova a caricare NULL nella colonna.

Se COPY tenta di assegnare NULL a una colonna definita come NOT NULL, il comando COPY non viene eseguito.

COPY mappa automaticamente gli elementi di dati nei dati di origine JSON alle colonne della tabella di destinazione. Lo fa associando chiavi di oggetti, o nomi, nelle coppie nome-valore di origine ai nomi delle colonne nella tabella di destinazione.

Fare riferimento ai seguenti dettagli su ogni valore json_option:

'auto'

> Questa opzione fa distinzione tra lettere maiuscole e minuscole. I nomi delle colonne nelle tabelle Amazon Redshift sono sempre in minuscolo, quindi quando si utilizza l'opzione `'auto'`, anche i nomi dei campi JSON corrispondenti devono essere in minuscolo.

'auto ignorecase'

Con questa opzione, non si fa distinzione tra maiuscole e minuscole. I nomi delle colonne nelle tabelle Amazon Redshift sono sempre minuscoli, quindi quando si utilizza l'opzione `'auto ignorecase'`, i nomi dei campi JSON corrispondenti possono essere minuscoli, maiuscoli o misti.

's3://jsonpaths_file'

Con questa opzione, COPY utilizza il file JSONPaths per mappare gli elementi dei dati nei dati di origine JSON sulle colonne della tabella di destinazione. L'argomento *`s3://jsonpaths_file`* deve essere una chiave oggetto Amazon S3 che fa esplicito riferimento a un singolo file. Un esempio è `'s3://mybucket/jsonpaths.txt'`. L'argomento non può essere un prefisso di chiave. Per ulteriori informazioni sull'utilizzo di un file JSONPath, consultare the section called “File JSONPath”.

In alcuni casi, il file specificato da `jsonpaths_file` ha lo stesso prefisso del percorso specificato da `copy_from_s3_objectpath` per i file di dati. In tal caso, COPY legge il file JSONPaths come file di dati e restituisce errori. Ad esempio, si supponga che i file di dati utilizzino il percorso dell'oggetto `s3://mybucket/my_data.json` e che il file JSONPaths sia `s3://mybucket/my_data.jsonpaths`. In questo caso, COPY prova a caricare `my_data.jsonpaths` come file di dati.

'noshred'

Con questa opzione, Amazon Redshift non divide gli attributi delle strutture JSON in più colonne durante il caricamento di un documento JSON.

File di dati JSON

Il file di dati JSON contiene un insieme di oggetti o array. COPY carica ogni oggetto o array JSON in una riga della tabella di destinazione. Ogni oggetto o array corrispondente a una riga deve essere una struttura autonoma a livello di root, cioè non deve essere membro di un'altra struttura JSON.

Un oggetto JSON inizia e termina con delle parentesi ({ }) e contiene un insieme non ordinato di coppie nome-valore. Ogni coppia nome e valore è separata da due punti e le coppie sono separate da virgole. Per impostazione predefinita, la chiave oggetto, o nome, nelle coppie nome-valore deve corrispondere al nome della colonna corrispondente nella tabella. I nomi delle colonne nelle tabelle Amazon Redshift sono sempre in minuscolo, quindi anche le chiavi dei nomi dei campi JSON corrispondenti devono essere in minuscolo. Se i nomi delle colonne e le chiavi JSON non

corrispondono, utilizza un the section called “File JSONPath” per mappare esplicitamente le colonne sulle chiavi.

L'ordine non ha importanza in un oggetto JSON. I nomi che non corrispondono al nome di una colonna vengono ignorati. Di seguito viene mostrata la struttura di un semplice oggetto JSON.

```
{
  "column1": "value1",
  "column2": value2,
  "notacolumn" : "ignore this value"
}
```

Un array JSON inizia e termina con delle parentesi ({ }) e contiene un insieme ordinato di valori separati da virgole. Se i file di dati utilizzano array, è necessario specificare un file JSONPath per far corrispondere i valori alle colonne. Di seguito viene mostrata la struttura di un semplice array JSON.

```
["value1", value2]
```

Il JSON deve essere ben formato. Ad esempio, gli oggetti o gli array non possono essere separati da virgole o altri caratteri tranne lo spazio. Le stringhe devono essere racchiuse tra virgolette doppie. I caratteri virgolette devono essere virgolette semplici (0x22), non oblique o "intelligenti".

La dimensione massima di un singolo oggetto o array JSON, compresi parentesi graffe o quadre, è di 4 MB. È diverso dalla dimensione massima della riga. Se la dimensione massima di un singolo oggetto o array JSON viene superata, anche se la dimensione della riga risultante è inferiore al limite di 4 MB, il comando COPY non viene eseguito.

Nel calcolo delle dimensioni delle righe, Amazon Redshift internamente conta due volte i caratteri pipe (|). Se i dati immessi contengono un numero molto elevato di caratteri pipe, è possibile che la dimensione delle righe superi i 4 MB anche se la dimensione dell'oggetto è inferiore a 4 MB.

COPY carica `\n` come carattere newline e carica `\t` come carattere di tabulazione. Per caricare una barra rovesciata, crea una sequenza di escape con una barra rovesciata (`\\`).

COPY cerca il sorgente JSON specificato per un oggetto o array JSON valido e ben strutturato. Se COPY rileva caratteri non vuoti prima di individuare una struttura JSON utilizzabile o tra oggetti o array JSON validi, COPY restituisce un errore per ogni istanza. Questi errori contano ai fini del conteggio MAXERROR. Quando il conteggio degli errori è uguale o superiore a MAXERROR, COPY fallisce.

Per ogni errore, Amazon Redshift registra una riga nella tabella di sistema STL_LOAD_ERRORS. La colonna LINE_NUMBER registra l'ultima riga dell'oggetto JSON che ha causato l'errore.

Se è specificato IGNOREHEADER, COPY ignora il numero di righe specificato nei dati JSON. I caratteri newline nei dati JSON sono sempre conteggiati per i calcoli IGNOREHEADER.

COPY carica le stringhe vuote come campi vuoti per impostazione predefinita. Se è specificato EMPTYASNULL, COPY carica le stringhe vuote per i campi CHAR e VARCHAR come NULL. Le stringhe vuote per altri tipi di dati, come ad esempio INT, vengono sempre caricate come NULL.

Con JSON non sono supportate le seguenti opzioni:

- CSV
- DELIMITER
- ESCAPE
- FILLRECORD
- FIXEDWIDTH
- IGNOREBLANKLINES
- NULL AS
- READRATIO
- REMOVEQUOTES

Per ulteriori informazioni, consulta COPY dal formato JSON. Per ulteriori informazioni sulle strutture dati JSON, visita il sito http://www.json.org/.

File JSONPath

Se si carica da dati di origine con formattazione JSON o Avro, per impostazione predefinita COPY mappa gli elementi di dati di primo livello nei dati di origine alle colonne della tabella di destinazione. Lo fa associando ogni nome, o chiave di oggetti, in una coppia nome-valore al nome di una colonna nella tabella di destinazione.

Se i nomi delle colonne e delle chiavi oggetto non corrispondono o per mappare a livelli più profondi nella gerarchia dei dati, è possibile utilizzare un file JSONPath per mappare esplicitamente gli elementi di dati JSON o Avro sulle colonne. Il file JSONPath mappa gli elementi di dati JSON sulle colonne facendo corrispondere l'ordine delle colonne nella tabella di destinazione o nell'elenco delle colonne.

Il file JSONPath deve contenere solo un singolo oggetto JSON (non un array). L'oggetto JSON è una coppia nome-valore. La chiave oggetto, che è il nome nella coppia nome-valore, deve essere `"jsonpaths"`. Il valore nella coppia nome-valore è un array di espressioni JSONPath. Ogni espressione JSONPath fa riferimento a un singolo elemento nella gerarchia dati JSON o nello schema Avro, analogamente a come un'espressione XPath fa riferimento agli elementi di un documento XML. Per ulteriori informazioni, consulta Espressioni JSONPath.

Per utilizzare un file JSONPaths, aggiungere la parola chiave JSON o AVRO al comando COPY. Specificare il nome del bucket S3 e il percorso dell'oggetto del file JSONPaths utilizzando il seguente formato.

```
COPY tablename
FROM 'data_source'
CREDENTIALS 'credentials-args'
FORMAT AS { AVRO | JSON } 's3://jsonpaths_file';
```

Il valore `s3://jsonpaths_file` deve essere una chiave oggetto di Amazon S3 che fa riferimento esplicitamente a singolo file, come `'s3://mybucket/jsonpaths.txt'`. Non può essere un prefisso di chiave.

In alcuni casi, se il caricamento avviene da Amazon S3 il file specificato da `jsonpaths_file` ha lo stesso prefisso del percorso specificato da `copy_from_s3_objectpath` per i file di dati. In tal caso, COPY legge il file JSONPaths come file di dati e restituisce errori. Ad esempio, si supponga che i file di dati utilizzino il percorso dell'oggetto `s3://mybucket/my_data.json` e che il file JSONPaths sia `s3://mybucket/my_data.jsonpaths`. In questo caso, COPY prova a caricare `my_data.jsonpaths` come file di dati.

Se il nome della chiave è una stringa diversa da `"jsonpaths"`, il comando COPY non restituisce un errore, ma ignora jsonpaths_file e usa invece l'argomento `'auto'`.

Se si verifica una delle seguenti situazioni, il comando COPY non va a buon fine:

- Il JSON ha un formato errato.
- C'è più di un oggetto JSON.
- Tutti i caratteri tranne lo spazio esistono al di fuori dell'oggetto.
- Un elemento array è una stringa vuota o non è una stringa.

MAXERROR non si applica al file JSONPath.

Il file JSONPath non deve essere crittato, anche se è specificata l'opzione ENCRYPTED.

Per ulteriori informazioni, consulta COPY dal formato JSON.

Espressioni JSONPath

Il file JSONPath utilizza le espressioni JSONPath per mappare i campi di dati sulle colonne di destinazione. Ogni espressione JSONPath corrisponde a una colonna nella tabella di destinazione di Amazon Redshift. L'ordine degli elementi dell'array JSONPath deve corrispondere all'ordine delle colonne della tabella di destinazione o, se viene utilizzato un elenco di colonne, di quest'ultimo.

I caratteri delle doppie virgolette sono richiesti come mostrato, sia per i nomi dei campi sia per i valori. I caratteri delle virgolette devono essere virgolette semplici (0x22), non oblique o "intelligenti".

Se un elemento oggetto a cui fa riferimento un'espressione JSONPath non viene trovato nei dati JSON, COPY tenta di caricare un valore NULL. Se l'oggetto di riferimento ha un formato errato, COPY restituisce un errore di caricamento.

Se un elemento array a cui fa riferimento un'espressione JSONPath non viene trovato nei dati JSON o Avro, COPY non va a buon fine e restituisce il seguente errore: `Invalid JSONPath format: Not an array or index out of range.` Rimuovi qualsiasi elemento array dal JSONPath che non esiste nei dati sorgente e verifica che gli array nei dati sorgente abbiano un formato corretto.

Le espressioni JSONPath possono utilizzare la notazione con parentesi o punti, ma non è possibile mischiare le notazioni. L'esempio seguente mostra le espressioni JSONPath che utilizzano la notazione con parentesi.

```
{
    "jsonpaths": [
        "$['venuename']",
        "$['venuecity']",
        "$['venuestate']",
        "$['venueseats']"
    ]
}
```

L'esempio seguente mostra le espressioni JSONPath che utilizzano la notazione con punti.

```
{
    "jsonpaths": [
```

```
        "$.venuename",
        "$.venuecity",
        "$.venuestate",
        "$.venueseats"
    ]
}
```

Nel contesto della sintassi di COPY di Amazon Redshift, un'espressione JSONPath deve specificare il percorso esplicito di un singolo elemento del nome in una struttura di dati gerarchica JSON o Avro. Amazon Redshift non supporta elementi JSONPath, come caratteri jolly o espressioni filtro, che potrebbero determinare un percorso ambiguo o elementi con più nomi.

Per ulteriori informazioni, consulta COPY dal formato JSON.

Utilizzo di JSONPath con dati Avro

L'esempio seguente mostra uno schema Avro con più livelli.

```
{
    "name": "person",
    "type": "record",
    "fields": [
        {"name": "id", "type": "int"},
        {"name": "guid", "type": "string"},
        {"name": "isActive", "type": "boolean"},
        {"name": "age", "type": "int"},
        {"name": "name", "type": "string"},
        {"name": "address", "type": "string"},
        {"name": "latitude", "type": "double"},
        {"name": "longitude", "type": "double"},
        {
            "name": "tags",
            "type": {
                        "type" : "array",
                        "name" : "inner_tags",
                        "items" : "string"
                    }
        },
        {
            "name": "friends",
            "type": {
                        "type" : "array",
                        "name" : "inner_friends",
```

```
                        "items" : {
                                    "name" : "friends_record",
                                    "type" : "record",
                                    "fields" : [
                                                {"name" : "id", "type" : "int"},
                                                {"name" : "name", "type" : "string"}
                                               ]
                                   }
                    }
        },
        {"name": "randomArrayItem", "type": "string"}
    ]
}
```

L'esempio seguente mostra un file JsonPaths che utilizza AvroPath espressioni per fare riferimento allo schema precedente.

```
{
    "jsonpaths": [
        "$.id",
        "$.guid",
        "$.address",
        "$.friends[0].id"
    ]
}
```

L'esempio di JSONPath include i seguenti elementi:

jsonpath

Il nome dell'oggetto JSON che contiene le espressioni. AvroPath

[...]

Le parentesi racchiudono l'array JSON che contiene gli elementi del percorso.

$

Il segno del dollaro fa riferimento all'elemento root nello schema Avro, che è l'array `"fields"`.

"$.id",

La destinazione dell' AvroPath espressione. In questa istanza, la destinazione è l'elemento nell'array `"fields"` con il nome `"id"`. Le espressioni sono separate da virgole.

"$.friends[0].id"

Le parentesi indicano un indice dell'array. Le espressioni JSONPath utilizzano un'indicizzazione a base zero, quindi questa espressione fa riferimento al primo elemento nell'array `"friends"` con il nome `"id"`.

La sintassi dello schema Avro richiede l'utilizzo di campi interni per definire la struttura dei tipi di dati record e array. I campi interni vengono ignorati dalle AvroPath espressioni. Per esempio, il campo `"friends"` definisce un array chiamato `"inner_friends"`, che a sua volta definisce un record chiamato `"friends_record"`. L' AvroPath espressione che fa riferimento al campo `"id"` può ignorare i campi aggiuntivi per fare riferimento direttamente al campo di destinazione. Le AvroPath espressioni seguenti fanno riferimento ai due campi che appartengono all'`"friends"`array.

```
"$.friends[0].id"
"$.friends[0].name"
```

Parametri del formato dei dati a colonna

Oltre ai formati di dati standard, COPY supporta i seguenti formati di dati a colonna per COPY da Amazon S3: COPY dal formato a colonne è supportato con alcune restrizioni. Per ulteriori informazioni, consulta COPY da formati di dati a colonna.

ORC

Carica i dati da un file che utilizza il formato di file ORC (Optimized Row Columnar).

PARQUET

Carica i dati da un file che utilizza il formato di file Parquet.

Parametri di compressione dei file

È possibile caricare da file di dati compressi specificando i seguenti parametri.

Parametri di compressione dei file

BZIP2

Un valore che specifica che il file o i file di input sono nel formato compresso bzip2 (file .bz2). L'operazione COPY legge ogni file compresso e decomprime i dati man mano che vengono caricati.

GZIP

Un valore che specifica che il file o i file di input sono nel formato compresso gzip (file .gz). L'operazione COPY legge ogni file compresso e decomprime i dati man mano che vengono caricati.

LZOP

Un valore che specifica che il file o i file di input sono nel formato compresso lzop (file .lzo). L'operazione COPY legge ogni file compresso e decomprime i dati man mano che vengono caricati.

> **Note**
>
> COPY non supporta i file che vengono compressi usando l'opzione --filter di lzop.

ZSTD

Un valore che specifica che il file o i file di input sono nel formato compresso Zstandard (file .zst). L'operazione COPY legge ogni file compresso e decomprime i dati man mano che vengono caricati. Per ulteriori informazioni, consulta ZSTD.

> **Note**
>
> ZSTD è supportato solo con COPY da Amazon S3.

Parametri di conversione dei dati

Mentre carica la tabella, COPY tenta di convertire in modo implicito le stringhe nei dati di origine nel tipo di dati della colonna di destinazione. Se hai necessità di specificare una conversione diversa dal comportamento predefinito o se la conversione predefinita dà luogo a errori, è possibile gestire le conversioni dei dati specificando i seguenti parametri. Per ulteriori informazioni sulla sintassi dei parametri, consulta Sintassi di COPY.

- ACCEPTANYDATE
- ACCEPTINVCHARS
- BLANKSASNULL

- DATEFORMAT
- EMPTYASNULL
- ENCODING
- ESCAPE
- EXPLICIT_IDS
- FILLRECORD
- IGNOREBLANKLINES
- IGNOREHEADER
- NULL AS
- REMOVEQUOTES
- ROUNDEC
- TIMEFORMAT
- TRIMBLANKS
- TRUNCATECOLUMNS

Parametri di conversione dei dati

ACCEPTANYDATE

Consente il caricamento di qualsiasi formato di data, compresi i formati non validi come `00/00/00 00:00:00` senza generare un errore. Questo parametro è valido solo alle colonne TIMESTAMP e DATE. Utilizza sempre ACCEPTANYDATE con il parametro DATEFORMAT. Se il formato della data per i dati non corrisponde alla specifica DATEFORMAT, Amazon Redshift inserisce un valore NULL nel campo.

ACCEPTINVCHARS [AS] ['replacement_char']

Consente il caricamento di dati nelle colonne VARCHAR anche se i dati contengono caratteri UTF-8 non validi. Quando è specificato ACCEPTINVCHARS, COPY sostituisce ogni carattere UTF-8 non valido con una stringa di uguale lunghezza composta dal carattere specificato da replacement_char. Ad esempio, se il carattere sostitutivo è "^", un carattere a tre byte non valido verrà sostituito con "^^^".

Il carattere sostitutivo può essere qualsiasi carattere ASCII tranne NULL. Il valore predefinito è un punto interrogativo (?). Per informazioni sui caratteri UTF-8 non validi, consultare Errori di caricamento di caratteri multibyte.

COPY restituisce il numero di righe che contenevano caratteri UTF-8 non validi e aggiunge una voce alla tabella di sistema STL_REPLACEMENTS per ogni riga interessata, fino a un massimo di 100 righe per ogni porzione di nodo. Vengono sostituiti anche altri caratteri UTF-8 non validi, ma gli eventi di sostituzione non vengono registrati.

Se non è specificato ACCEPTINVCHARS, COPY restituisce un errore ogni volta che incontra un carattere UTF-8 non valido.

ACCEPTINVCHARS è valido solo per le colonne VARCHAR.

BLANKSASNULL

Carica come NULL i campi vuoti, composti solo da caratteri di spazio. Questa opzione è valide solo alle colonne CHAR e VARCHAR. I campi vuoti per altri tipi di dati, come ad esempio INT, vengono sempre caricati come NULL. Ad esempio, una stringa che contiene tre caratteri spazio in successione (e nessun altro carattere) viene caricata come NULL. Il comportamento predefinito, senza questa opzione, è di caricare i caratteri spazio così come sono.

DATEFORMAT [AS] {'dateformat_string' | 'auto' }

Se non è specificato nessun DATEFORMAT, il formato predefinito è `'YYYY-MM-DD'`. Ad esempio, un formato valido alternativo è `'MM-DD-YYYY'`.

Se il comando COPY non riconosce il formato dei valori di data e ora o se i valori di data e ora utilizzano formati diversi, utilizza l'argomento `'auto'` con il parametro DATEFORMAT o TIMEFORMAT. L'argomento `'auto'` riconosce diversi formati che non sono supportati quando si usa una stringa DATEFORMAT e TIMEFORMAT. La parola chiave `'auto''` prevede una distinzione tra lettere maiuscole e minuscole. Per ulteriori informazioni, consulta Utilizzo del riconoscimento automatico con DATEFORMAT e TIMEFORMAT.

Il formato della data può includere informazioni sull'ora (ore, minuti, secondi), ma tali informazioni vengono ignorate. La parola chiave AS è facoltativa. Per ulteriori informazioni, consulta Stringhe DATEFORMAT e TIMEFORMAT.

EMPTYASNULL

Indica che Amazon Redshift dovrebbe caricare i campi vuoti CHAR e VARCHAR come NULL. I campi vuoti per altri tipi di dati, come ad esempio INT, vengono sempre caricati come NULL. I campi vuoti si hanno quando i dati contengono due delimitatori in successione senza caratteri tra i delimitatori. EMPTYASNULL e NULL AS '' (stringa vuota) producono lo stesso comportamento.

ENCODING [AS] file_encoding

Specifica il tipo di codifica dei dati di caricamento. Il comando COPY converte i dati dalla codifica specificata a UTF-8 durante il caricamento.

I valori validi per file_encoding sono i seguenti:

- `UTF8`
- `UTF16`
- `UTF16LE`
- `UTF16BE`

Il valore predefinito è `UTF8`.

I nomi dei file di origine devono utilizzare la codifica UTF-8.

I seguenti file devono utilizzare la codifica UTF-8, anche se per i dati di caricamento è specificata una codifica diversa:

- File manifesto
- File JSONPath

Le stringhe di argomento con i seguenti parametri devono utilizzare UTF-8:

- FIXEDWIDTH 'fixedwidth_spec'
- ACCEPTINVCHARS 'replacement_char'
- DATEFORMAT 'dateformat_string'
- TIMEFORMAT 'timeformat_string'
- NULL AS 'null_string'

I file di dati a larghezza fissa devono utilizzare la codifica UTF-8. Le larghezze dei campi si basano sul numero di caratteri, non sul numero di byte.

Tutti i dati di caricamento devono utilizzare la codifica specificata. Se COPY incontra una codifica diversa, salta il file e restituisce un errore.

Se specifichi `UTF16`, i dati devono avere un BOM (byte order mark). Se sai se i dati UTF-16 sono little-endian (LE) o big-endian (BE), è possibile utilizzare `UTF16LE` o `UTF16BE` indipendentemente dalla presenza di un BOM.

ESCAPE

Quando si specifica questo parametro, il carattere di barra rovesciata (\) nei dati di input viene trattato come un carattere di escape. Il carattere che segue immediatamente il carattere di barra rovesciata viene caricato nella tabella come parte del valore corrente della colonna, anche se è un carattere che normalmente ha uno scopo particolare. Ad esempio, è possibile utilizzare questo parametro per creare una sequenza di escape con il carattere delimitatore, un punto interrogativo, un carattere newline incorporato o il carattere di escape stesso quando uno di questi caratteri è una parte lecita di un valore di colonna.

Se specifichi il parametro ESCAPE in combinazione con il parametro REMOVEQUOTES, è possibile creare una sequenza di escape e mantenere le virgolette (' o '') che altrimenti potrebbero essere rimosse. La stringa nulla predefinita, `\N`, funziona così com'è, ma è possibile anche creare una sequenza di escape nei dati di input come `\\N`. Finché non specifichi una stringa nulla alternativa con il parametro NULL AS, `\N` e `\\N` produrranno gli stessi risultati.

> **Note**
>
> Non è possibile eseguire l'escape del carattere di controllo `0x00` (NUL) e questo carattere deve essere rimosso dai dati di input o convertito. Questo carattere viene trattato come un marcatore EOR (end of record), causando il troncamento del resto del record.

Non è possibile utilizzare il parametro ESCAPE per caricamenti FIXEDWIDTH e non è possibile specificare il carattere di escape stesso; il carattere di escape è sempre il carattere di barra rovesciata. Inoltre, devi assicurarti che i dati di input contengano il carattere di escape nei punti appropriati.

Di seguito sono riportati alcuni esempi di dati di input e i dati caricati risultanti quando viene specificato il parametro ESCAPE. Il risultato per la riga 4 presuppone che sia stato specificato anche il parametro REMOVEQUOTES. I dati di input sono costituiti da due campi delimitati da pipe:

```
1|The quick brown fox\[newline]
jumped over the lazy dog.
2| A\\B\\C
3| A \| B \| C
4| 'A Midsummer Night\'s Dream'
```

I dati caricati nella colonna 2 assomigliano a questi:

```
The quick brown fox
jumped over the lazy dog.
A\B\C
A|B|C
A Midsummer Night's Dream
```

> **Note**
>
> L'applicazione del carattere di escape ai dati di input per un caricamento è responsabilità dell'utente. Un'eccezione a questo requisito è quando ricarichi dati che erano stati precedentemente scaricati con il parametro ESCAPE. In questo caso, i dati conterranno già i caratteri di escape necessari.

Il parametro ESCAPE non interpreta la notazione ottale, esadecimale, Unicode o altra sequenza di escape. Ad esempio, se i dati di origine contengono il valore di avanzamento della linea ottale (`\012`) e si prova a caricare questi dati con il parametro ESCAPE, Amazon Redshift carica il valore `012` nella tabella e non interpreta questo valore come un avanzamento di linea in una sequenza di escape.

Per inserire in una sequenza di escape i caratteri newline nei dati che provengono dalle piattaforme Microsoft Windows, potresti dover utilizzare due caratteri di escape: uno per il ritorno a capo e uno per l'avanzamento di linea. In alternativa, è possibile rimuovere il ritorno a capo prima di caricare il file (ad esempio, utilizzando l'utilità dos2unix).

EXPLICIT_IDS

Utilizza EXPLICIT_IDS con tabelle che hanno colonne IDENTITY se desideri sostituire i valori generati automaticamente con valori espliciti dai file di dati sorgente per le tabelle. Se il comando include un elenco di colonne, tale elenco deve includere le colonne IDENTITY per poter utilizzare questo parametro. Il formato dei dati per i valori di EXPLICIT_IDS deve corrispondere al formato IDENTITY specificato dalla definizione CREATE TABLE.

Quando esegui un comando COPY su una tabella con l'opzione EXPLICIT_IDS, Amazon Redshift non controlla più l'univocità delle colonne IDENTITY della tabella.

Se una colonna è definita con GENERATED BY DEFAULT AS IDENTITY, può esser copiata. i valori vengono generati o aggiornati con i valori forniti dall'utente. L'opzione EXPLICIT_IDS non è obbligatoria. COPY non aggiorna la filigrana ad elevata identità.

Per un esempio di comando COPY con EXPLICIT_IDS, consulta Caricamento di VENUE con valori espliciti per una colonna IDENTITY.

FILLRECORD

Consente di caricare file di dati quando mancano colonne contigue alla fine di alcuni record. Le colonne mancanti vengono caricate come NULL. Per i formati testo e CSV, se la colonna mancante è una colonna VARCHAR, vengono caricate stringhe a lunghezza zero anziché NULL. Per caricare NULL nelle colonne VARCHAR dal testo e dal CSV, specificare la parola chiave EMPTYASNULL. La sostituzione NULL funziona solo se la definizione della colonna permette dei NULL.

Per esempio, se la definizione della tabella contiene quattro colonne CHAR che possono contenere dei null e un record contiene i valori `apple, orange, banana, mango`, il comando COPY può caricare e compilare un record che contiene solo i valori `apple, orange`. I valori CHAR mancanti saranno caricati come valori NULL.

IGNOREBLANKLINES

Ignora le righe vuote che contengono solo un avanzamento di linea in un file di dati e non tenta di caricarle.

IGNOREHEADER [AS] number_rows

Tratta il number_rows specificato come intestazione del file e non carica le righe. Usa IGNOREHEADER per saltare le intestazioni di tutti i file in un caricamento parallelo.

NULL AS 'null_string'

Carica i campi che fanno corrispondere null_string a NULL, dove null_string può essere una stringa qualsiasi. Se i dati includono un terminatore null, chiamato anche NUL (UTF-8 0000) o zero binario (0x000), COPY lo tratta come qualsiasi altro carattere. Ad esempio, un record contenente '1' || NUL || '2' viene copiato come stringa di lunghezza da 3 byte. Se un campo contiene solo NUL, è possibile utilizzare NULL AS per sostituire il terminatore null con NULL specificando `'\0'` o `'\000'`, ad esempio `NULL AS '\0'` o `NULL AS '\000'`. Se un campo contiene una stringa che termina con NUL ed è specificato NULL AS, la stringa viene inserita con NUL alla fine. Non utilizzare '\n' (newline) per il valore null_string. Amazon Redshift riserva '\n' per l'utilizzo come delimitatore di linea. La null_string predefinita è `'\N'`.

> **Note**
>
> Se tenti di caricare i nulli in una colonna definita come NON NULL, il comando COPY fallirà.

REMOVEQUOTES

Rimuove le virgolette intorno alle stringhe nei dati in entrata. Tutti i caratteri compresi tra le virgolette, inclusi i delimitatori, vengono mantenuti. Se una stringa ha una virgoletta iniziale singola o doppia ma non una virgoletta finale corrispondente, il comando COPY non carica quella riga e restituisce un errore. La seguente tabella mostra alcuni semplici esempi di stringhe che contengono virgolette e i valori caricati risultanti.

Stringa di input	Valore caricato con opzione REMOVEQUOTES
"Il delimitatore è un carattere pipe (|)"	Il delimitatore è un carattere pipe (|)
'Nero'	Nero
"Bianco"	Bianco
Blu'	Blu'
'Blu	Valore non caricato: condizione di errore
"Blu	Valore non caricato: condizione di errore
' ' 'Nero' ' '	' 'Nero' '
' '	<white space>

ROUNDEC

Arrotonda i valori numerici quando la scala del valore di input è maggiore della scala della colonna. Per impostazione predefinita, COPY tronca i valori quando necessario per adattarli alla scala della colonna. Per esempio, se un valore di `20.259` viene caricato in una colonna DECIMAL(8,2), COPY tronca il valore a `20.25` per impostazione predefinita. Se è specificato

ROUNDEC, COPY arrotonda il valore a `20.26`. Il comando INSERT arrotonda sempre i valori quando necessario per farli corrispondere alla scala della colonna, quindi un comando COPY con il parametro ROUNDEC si comporta come un comando INSERT.

TIMEFORMAT [AS] {'timeformat_string' | 'auto' | 'epochsecs' | 'epochmillisecs' }

Specifica il formato dell'ora. Se non è specificato alcun TIMEFORMAT, il formato predefinito è `YYYY-MM-DD HH:MI:SS` per le colonne TIMESTAMP o `YYYY-MM-DD HH:MI:SSOF` per le colonne TIMESTAMPTZ, dove `OF` è l'offset rispetto all'UTC (Tempo coordinato universale). Non è possibile includere uno specificatore di fuso orario in timeformat_string. Per caricare i dati TIMESTAMPTZ in un formato diverso da quello predefinito, specificare "auto"; per maggiori informazioni, consultare Utilizzo del riconoscimento automatico con DATEFORMAT e TIMEFORMAT. Per maggiori informazioni su timeformat_string, consultare Stringhe DATEFORMAT e TIMEFORMAT.

L'argomento `'auto'` riconosce diversi formati che non sono supportati quando si usa una stringa DATEFORMAT e TIMEFORMAT. Se il comando COPY non riconosce il formato dei valori di data e ora o se i valori di data e ora utilizzano formati diversi uno dall'altro, utilizza l'argomento `'auto'` con il parametro DATEFORMAT o TIMEFORMAT. Per ulteriori informazioni, consulta Utilizzo del riconoscimento automatico con DATEFORMAT e TIMEFORMAT.

Se i dati sorgente sono rappresentati come tempo dell'epoca, cioè il numero di secondi o millisecondi dal 1° gennaio 1970, 00:00:00 UTC, specifica `'epochsecs'` o `'epochmillisecs'`.

Le parole chiave `'auto'`, `'epochsecs'` e `'epochmillisecs'` prevedono una distinzione tra lettere maiuscole e minuscole.

La parola chiave AS è facoltativa.

TRIMBLANKS

Rimuove i caratteri di spazio finale da una stringa VARCHAR. Questo parametro è valido solo per le colonne con un tipo di dati VARCHAR.

TRUNCATECOLUMNS

Tronca i dati nelle colonne al numero appropriato di caratteri in modo che corrispondano alle specifiche della colonna. Si applica solo alle colonne con un tipo di dati VARCHAR o CHAR e alle righe di dimensioni fino a 4 MB.

Operazioni di caricamento dati

Gestisce il comportamento predefinito dell'operazione di caricamento per la risoluzione dei problemi o per ridurre i tempi di caricamento specificando i seguenti parametri.

- COMPROWS
- COMPUPDATE
- IGNOREALLERRORS
- MAXERROR
- NOLOAD
- STATUPDATE

Parametri

COMPROWS numrows

Specifica il numero di righe da utilizzare come dimensione del campione per l'analisi della compressione. L'analisi viene eseguita su righe da ciascuna sezione di dati. Ad esempio, se specifichi `COMPROWS 1000000` (1,000,000) e il sistema contiene quattro sezioni totali, vengono lette e analizzate non più di 250.000 righe per ogni sezione.

Se COMPROWS non è specificato, la dimensione del campione è impostata su 100.000 per ogni sezione. I valori di COMPROWS inferiori al valore predefinito di 100.000 righe per ogni sezione vengono automaticamente aggiornati al valore predefinito. Tuttavia, la compressione automatica non verrà eseguita se la quantità di dati caricati è insufficiente per produrre un campione significativo.

Se il numero di COMPROWS è maggiore del numero di righe del file di input, il comando COPY continua ad eseguire l'analisi di compressione su tutte le righe disponibili. L'intervallo accettato per questo argomento è un numero compreso tra 1000 e 2147483647 (2.147.483.647).

COMPUPDATE [PRESET | { ON | TRUE } | { OFF | FALSE }],

Controlla se le codifiche di compressione vengono applicate automaticamente durante un COPY.

Quando COMPUPDATE è PRESET, il comando COPY sceglie la codifica di compressione per ogni colonna se la tabella di destinazione è vuota; anche se le colonne hanno già codifiche diverse da RAW. Le codifiche di colonna specificate attualmente possono essere sostituite. La

codifica per ogni colonna si basa sul tipo di dati della colonna. Nessun dato viene campionato. Amazon Redshift assegna automaticamente la codifica della compressione come segue:

- Le colonne definite come chiavi di ordinamento vengono assegnate alla compressione RAW.
- Le colonne definite come tipi di dati BOOLEAN, REAL o DOUBLE PRECISION vengono assegnate alla compressione RAW.
- Le colonne definite come SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIMESTAMP o TIMESTAMPTZ sono assegnate alla compressione AZ64.
- Le colonne definite come CHAR o VARCHAR sono assegnate alla compressione LZO.

Quando COMPUPDATE viene omesso, il comando COPY sceglie la codifica di compressione per ciascuna colonna solo se la tabella di destinazione è vuota e non è stata specificata una codifica (diversa da RAW) per nessuna colonna. La codifica per ciascuna colonna è determinata da Amazon Redshift. Nessun dato viene campionato.

Quando COMPUPDATE è impostato su ON (o TRUE) o COMPUPDATE viene specificato senza un'opzione, il comando COPY applica la compressione automatica se la tabella è vuota, anche se le colonne della tabella possiedono già codifiche diverse da RAW. Le codifiche di colonna specificate attualmente possono essere sostituite. La codifica di ciascuna colonna è basata su un'analisi dei dati campione. Per ulteriori informazioni, consulta Caricamento di tabelle con compressione automatica.

Con COMPUPDATE è impostato su OFF (o FALSE), la compressione automatica è disabilitata. Le codifiche di colonna non vengono modificate.

Per informazioni sulla tabella di sistema per analizzare la compressione, consultare STL_ANALYZE_COMPRESSION.

IGNOREALLERRORS

Puoi specificare questa opzione per ignorare tutti gli errori che si verificano durante l'operazione di caricamento.

Non è possibile specificare l'opzione IGNOREALLERRORS se si specifica l'opzione MAXERROR. Non è possibile specificare l'opzione IGNOREALLERRORS per i formati in colonna tra cui ORC e Parquet.

MAXERROR [AS] error_count

Se il caricamento restituisce il error_count numero di errori o superiore, il caricamento fallisce. Se il caricamento restituisce meno errori, continua e restituisce un messaggio INFO che indica il numero di righe che non è stato possibile caricare. Utilizza questo parametro per permettere la

prosecuzione del caricamento quando alcune righe non vengono caricate nella tabella a causa di errori di formattazione o altre incongruenze nei dati.

Imposta questo valore su `0` o `1` se desideri che il caricamento fallisca non appena si verifica il primo errore. La parola chiave AS è facoltativa. Il valore predefinito per MAXERROR è `0` e il limite è `100000`.

Il numero effettivo di errori segnalati potrebbe essere maggiore del MAXERROR specificato a causa della natura parallela di Amazon Redshift. Se un nodo qualsiasi nel cluster Amazon Redshift rileva che MAXERROR è stato superato, ogni nodo riporta tutti gli errori incontrati.

NOLOAD

Controlla la validità del file di dati senza caricare effettivamente i dati. Utilizza il parametro NOLOAD per assicurarti che il file di dati venga caricato senza errori prima di eseguire il caricamento effettivo dei dati. Eseguire COPY con il parametro NOLOAD è molto più veloce che caricare i dati, perché analizza solo i file.

STATUPDATE [{ ON | TRUE } | { OFF | FALSE }]

Regola il calcolo automatico e l'aggiornamento delle statistiche dell'ottimizzatore al termine di un comando COPY che ha avuto successo. Per impostazione predefinita, se il parametro STATUPDATE non viene utilizzato, le statistiche vengono aggiornate automaticamente se la tabella è inizialmente vuota.

Ogni volta che l'inserimento di dati in una tabella non vuota modifica significativamente le dimensioni della tabella, consigliamo di aggiornare le statistiche eseguendo un comando ANALYZE o utilizzando l'argomento STATUPDATE ON.

Con STATUPDATE ON (o TRUE), le statistiche vengono aggiornate automaticamente indipendentemente dal fatto che la tabella sia inizialmente vuota. Se si utilizza STATUPDATE, l'utente corrente deve essere il proprietario della tabella o un utente con privilegi avanzati. Se STATUPDATE non è specificato, è richiesta solo l'autorizzazione di INSERT.

Con STATUPDATE OFF (o FALSE), le statistiche non vengono mai aggiornate.

Per ulteriori informazioni, consultare Analisi delle tabelle.

Elenco alfabetico dei parametri

Il seguente elenco fornisce i collegamenti a ogni descrizione del parametro del comando COPY, in ordine alfabetico.

- ACCEPTANYDATE
- ACCEPTINVCHARS
- ACCESS_KEY_ID and SECRET_ACCESS_KEY
- AVRO
- BLANKSASNULL
- BZIP2
- COMPROWS
- COMPUPDATE
- CREDENTIALS
- CSV
- DATEFORMAT
- DELIMITER
- EMPTYASNULL
- ENCODING
- ENCRYPTED
- ESCAPE
- EXPLICIT_IDS
- FILLRECORD
- FIXEDWIDTH
- FORMAT
- FROM
- GZIP
- IAM_ROLE
- IGNOREALLERRORS
- IGNOREBLANKLINES
- IGNOREHEADER
- JSON
- LZOP
- MANIFEST
- MASTER_SYMMETRIC_KEY

- MAXERROR
- NOLOAD
- NULL AS
- READRATIO
- REGION
- REMOVEQUOTES
- ROUNDEC
- SESSION_TOKEN
- SHAPEFILE
- SSH
- STATUPDATE
- TIMEFORMAT
- SESSION_TOKEN
- TRIMBLANKS
- TRUNCATECOLUMNS
- ZSTD

Note per l'utilizzo

Argomenti

- Autorizzazioni per accedere ad altre risorse AWS
- Utilizzo di COPY con gli alias degli Access Point Amazon S3
- Caricamento di dati multibyte da Amazon S3
- Caricamento di una colonna definita come tipo dati GEOMETRY o GEOGRAPHY
- Caricamento del tipo di dati HLLSKETCH
- Caricamento di una colonna definita come tipo dati VARBYTE
- Errori durante la lettura di più file
- COPY dal formato JSON
- COPY da formati di dati a colonna
- Stringhe DATEFORMAT e TIMEFORMAT

- Utilizzo del riconoscimento automatico con DATEFORMAT e TIMEFORMAT

Autorizzazioni per accedere ad altre risorse AWS

Per spostare dati tra il cluster e un'altra AWS risorsa, come Amazon S3, Amazon DynamoDB, Amazon EMR o Amazon EC2, il cluster deve disporre dell'autorizzazione per accedere alla risorsa ed eseguire le azioni necessarie. Ad esempio, per caricare i dati da Amazon S3, COPY deve avere accesso LIST al bucket e accesso GET per gli oggetti del bucket. Per maggiori informazioni sulle autorizzazioni minime, consultare Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY.

Per ottenere l'autorizzazione per accedere alla risorsa, il cluster deve essere autenticato. È possibile scegliere uno dei seguenti metodi di autenticazione:

- Controllo degli accessi basato sui ruoli— Per il controllo degli accessi basato sui ruoli, è necessario specificare un ruolo AWS Identity and Access Management (IAM) utilizzato dal cluster per l'autenticazione e l'autorizzazione. Per proteggere AWS le credenziali e i dati sensibili, consigliamo vivamente di utilizzare l'autenticazione basata sui ruoli.
- Controllo degli accessi basato su chiave— Per il controllo degli accessi basato su chiavi, si forniscono le credenziali di AWS accesso (ID della chiave di accesso e chiave di accesso segreta) per un utente come testo semplice.

Controllo degli accessi basato sui ruoli

Con il controllo degli accessi basato su ruoli, il cluster assume in modo temporaneo e automatico un ruolo (IAM). Quindi, in base alle autorizzazioni concesse al ruolo, il cluster può accedere alle risorse AWS necessarie.

La creazione di un ruolo IAM è simile all'assegnazione delle autorizzazioni a un utente, in quanto è un'identità AWS con policy di autorizzazioni che determinano ciò che l'identità può e non può fare in AWS. Tuttavia, invece di essere associato univocamente a un utente, un ruolo può essere assunto da qualsiasi entità che ne abbia bisogno. Inoltre, a un ruolo non sono associate credenziali (password o chiavi di accesso). Al contrario, se un ruolo è associato a un cluster, le chiavi di accesso vengono create dinamicamente e fornite al cluster.

Ti consigliamo di utilizzare il controllo degli accessi basato sui ruoli perché offre un controllo più sicuro e dettagliato dell'accesso alle AWS risorse e ai dati sensibili degli utenti, oltre a salvaguardare le credenziali. AWS

L'autenticazione basata sui ruoli offre i seguenti vantaggi:

- Puoi utilizzare strumenti IAM AWS standard per definire un ruolo IAM e associarlo a più cluster. Quando modifichi la policy di accesso a un ruolo, le modifiche vengono applicate automaticamente a tutti i cluster che utilizzano il ruolo.
- Puoi definire policy IAM granulari che concedono autorizzazioni a cluster e utenti del database specifici per accedere a risorse e azioni specifiche. AWS
- Il cluster ottiene le credenziali di sessione temporanee al momento dell'esecuzione e le aggiorna come necessario fino al completamento dell'operazione. Se utilizzi credenziali temporanee basate su chiavi, l'operazione fallisce se le credenziali temporanee scadono prima del completamento.
- L'ID chiave di accesso e l'ID chiave di accesso segreta non vengono memorizzati o trasmessi nel codice SQL.

Per utilizzare il controllo degli accessi basato su ruoli, è necessario creare prima un ruolo IAM utilizzando il tipo di ruolo di servizio Amazon Redshift, quindi collegare il ruolo al cluster. Il ruolo deve avere, come minimo, le autorizzazioni elencate in Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY. Per i passaggi per creare un ruolo IAM e collegarlo al cluster, consulta la sezione Autorizzazione di Amazon Redshift ad accedere ad AWS altri servizi per conto dell'utente nella Amazon Redshift Management Guide.

È possibile aggiungere un ruolo a un cluster o visualizzare i ruoli associati a un cluster utilizzando la Console di gestione, la CLI o l'API di Amazon Redshift. Per ulteriori informazioni, consulta Associazione di un ruolo IAM a un cluster nella Guida alla gestione di Amazon Redshift.

Quando crei un ruolo IAM, IAM restituisce un Amazon Resource Name (ARN) per il ruolo. Per specificare un ruolo IAM, fornisci al ruolo ARN il parametro IAM_ROLE o il parametro CREDENTIALS.

Ad esempio, supponiamo che al cluster sia collegato il seguente ruolo.

```
"IamRoleArn": "arn:aws:iam::0123456789012:role/MyRedshiftRole"
```

Il seguente esempio di comando COPY utilizza il parametro IAM_ROLE con l'ARN dell'esempio precedente per l'autenticazione e l'accesso a Amazon S3.

```
copy customer from 's3://mybucket/mydata'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Il seguente esempio di comando COPY utilizza il parametro CREDENTIALS per specificare il ruolo IAM.

```
copy customer from 's3://mybucket/mydata'
credentials
'aws_iam_role=arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Inoltre, un utente con privilegi avanzati può concedere il privilegio ASSUMEROLE a utenti e gruppi di database per fornire l'accesso a un ruolo per le operazioni COPY. Per informazioni, consulta GRANT.

Controllo degli accessi basato su chiave

Con il controllo degli accessi basato su chiavi, fornisci l'ID della chiave di accesso e la chiave di accesso segreta per un utente IAM autorizzato ad accedere alle AWS risorse che contengono i dati. È possibile utilizzare sia i parametri ACCESS_KEY_ID and SECRET_ACCESS_KEY insieme, sia il parametro CREDENTIALS.

> **Note**
>
> Consigliamo vivamente di utilizzare un ruolo IAM per l'autenticazione invece di fornire un ID chiave di accesso in chiaro e una chiave di accesso segreta. Se scegli il controllo degli accessi basato su chiavi, non utilizzare mai le credenziali del tuo AWS account (root). Creare sempre un utente IAM e fornire l'ID della chiave di accesso e la chiave di accesso segreta di quell'utente. Per la procedura per creare un utente IAM, consultare Creazione di un utente IAM nell'account AWS.

Per l'autenticazione tramite ACCESS_KEY_ID e SECRET_ACCESS_KEY, sostituisci *<access-key-id>* e *<secret-access-key>* con un ID chiave di accesso di un utente autorizzato e una chiave di accesso completamente segreta, come mostrato di seguito.

```
ACCESS_KEY_ID '<access-key-id>'
SECRET_ACCESS_KEY '<secret-access-key>';
```

Per l'autenticazione tramite il parametro CREDENTIALS, sostituisci *<access-key-id>* e *<secret-access-key>* con un ID chiave di accesso di un utente autorizzato e una chiave di accesso completamente segreta, come mostrato di seguito.

```
CREDENTIALS
'aws_access_key_id=<access-key-id>;aws_secret_access_key=<secret-access-key>';
```

L'utente IAM deve disporre, come minimo, delle autorizzazioni elencate in Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY.

Credenziali di sicurezza temporanee

Se utilizzi il controllo degli accessi basato su chiave, è possibile limitare ulteriormente l'accesso ai dati degli utenti utilizzando le credenziali di sicurezza temporanee. L'autenticazione basata su ruoli utilizza automaticamente credenziali temporanee.

> **Note**
>
> Consigliamo vivamente di utilizzare role-based access control invece di creare credenziali temporanee e fornire ID chiave di accesso e chiave di accesso segreta come testo in chiaro. Il controllo degli accessi basato sui ruoli utilizza automaticamente le credenziali temporanee.

Le credenziali di sicurezza temporanee offrono maggiore sicurezza perché hanno una durata breve e non possono essere riutilizzate dopo la loro scadenza. L'ID chiave di accesso e la chiave di accesso segreta generate con il token non possono essere utilizzate senza il token e un utente che ha queste credenziali di sicurezza temporanee può accedere alle risorse solo fino a quando le credenziali non scadono.

Per concedere agli utenti l'accesso temporaneo alle tue risorse, chiami le operazioni API AWS Security Token Service (AWS STS). Le operazioni AWS STS API restituiscono credenziali di sicurezza temporanee costituite da un token di sicurezza, un ID della chiave di accesso e una chiave di accesso segreta. Rilascia le credenziali di sicurezza temporanee agli utenti che necessitano di un accesso temporaneo alle risorse. Questi utenti possono essere utenti IAM esistenti o utenti non AWS . Per ulteriori informazioni sulla creazione delle credenziali di sicurezza temporanee, consultare Utilizzo delle credenziali di sicurezza temporanee nella Guida per l'utente di IAM.

È possibile utilizzare sia i parametri ACCESS_KEY_ID and SECRET_ACCESS_KEY insieme con il parametro SESSION_TOKEN o il parametro CREDENTIALS. È inoltre necessario fornire l'ID chiave di accesso e la chiave di accesso segreta forniti con il token.

Per l'autenticazione tramite ACCESS_KEY_ID, SECRET_ACCESS_KEY e SESSION_TOKEN sostituisci *<temporary-access-key-id>*, *<temporary-secret-access-key>* e *<temporary-token>* come mostrato di seguito.

```
ACCESS_KEY_ID '<temporary-access-key-id>'
SECRET_ACCESS_KEY '<temporary-secret-access-key>'
```

```
SESSION_TOKEN '<temporary-token>';
```

Per autenticarti utilizzando CREDENTIALS, includi `session_token=<temporary-token>` nella stringa delle credenziali come mostrato di seguito.

```
CREDENTIALS
'aws_access_key_id=<temporary-access-key-id>;aws_secret_access_key=<temporary-secret-
access-key>;session_token=<temporary-token>';
```

L'esempio seguente mostra un comando COPY con credenziali di sicurezza temporanee.

```
copy table-name
from 's3://objectpath'
access_key_id '<temporary-access-key-id>'
secret_access_key '<temporary-secret-access-key>'
session_token '<temporary-token>';
```

L'esempio seguente carica la tabella LISTING con credenziali temporanee e crittografia dei file.

```
copy listing
from 's3://mybucket/data/listings_pipe.txt'
access_key_id '<temporary-access-key-id>'
secret_access_key '<temporary-secret-access-key>'
session_token '<temporary-token>'
master_symmetric_key '<root-key>'
encrypted;
```

L'esempio seguente carica la tabella LISTING utilizzando il parametro CREDENTIALS con credenziali temporanee e crittografia dei file.

```
copy listing
from 's3://mybucket/data/listings_pipe.txt'
credentials
'aws_access_key_id=<temporary-access-key-id>;aws_secret_access_key=<temporary-secret-
access-key>;session_token=<temporary-token>;master_symmetric_key=<root-key>'
encrypted;
```

> **Important**
>
> Le credenziali di sicurezza temporanee devono essere valide per l'intera durata dell'operazione COPY o UNLOAD. Se le credenziali di sicurezza temporanee scadono

durante l'operazione, il comando fallisce e la transazione viene annullata. Ad esempio, se le credenziali di sicurezza temporanee scadono dopo 15 minuti e l'operazione COPY richiede un'ora, l'operazione COPY fallisce prima del completamento. Se utilizzi l'accesso basato su ruoli, le credenziali di sicurezza temporanee vengono aggiornate automaticamente fino al completamento dell'operazione.

Autorizzazioni IAM per COPY, UNLOAD e CREATE LIBRARY

L'utente o il ruolo IAM a cui fa riferimento il parametro CREDENTIALS deve avere, come minimo, le seguenti autorizzazioni:

- Per COPY da Amazon S3, l'autorizzazione per il LIST del bucket Amazon S3 e il GET degli oggetti Amazon S3 che vengono caricati e il file manifest, se viene utilizzato.
- Per COPY da Amazon S3, Amazon EMR e host remoti (SSH) con dati formattati in JSON, l'autorizzazione per LIST e GET del file JSONPaths su Amazon S3, se viene utilizzato.
- Per COPY da DynamoDB, l'autorizzazione a SCAN e DESCRIBE sulla tabella DynamoDB che viene caricata.
- Per COPY da un cluster Amazon EMR, l'autorizzazione per l'operazione `ListInstances` sul cluster Amazon EMR.
- Per UNLOAD su Amazon S3, le autorizzazioni GET, LIST e PUT per il bucket Amazon S3 in cui i file di dati vengono scaricati.
- Per CREATE LIBRARY da Amazon S3, l'autorizzazione per il LIST del bucket Amazon S3 e il GET degli oggetti Amazon S3 importati.

> **Note**
>
> Se si riceve il messaggio di errore `S3ServiceException: Access Denied`, quando si esegue un comando COPY, UNLOAD o CREATE LIBRARY, il cluster non ha le autorizzazioni di accesso appropriate per Amazon S3.

È possibile gestire le autorizzazioni IAM assegnando una policy IAM a un ruolo IAM collegato al cluster, all'utente o al gruppo a cui appartiene l'utente. Ad esempio, la policy gestita da `AmazonS3ReadOnlyAccess` concede le autorizzazioni LIST e GET alle risorse Amazon S3. Per

ulteriori informazioni sulle policy IAM, consultare Utilizzo delle policy IAM nella Guida per l'utente di IAM.

Utilizzo di COPY con gli alias degli Access Point Amazon S3

Utilizzo di COPY con gli alias degli Access Point Amazon S3 Per ulteriori informazioni, consulta Utilizzo di un alias in stile bucket per il punto di accesso nella Guida utente di Amazon Simple Storage Service.

Caricamento di dati multibyte da Amazon S3

Se i dati includono caratteri multibyte non ASCII (ad esempio caratteri cinesi o cirillici), è necessario caricare i dati nelle colonne VARCHAR. Il tipo di dati VARCHAR supporta caratteri UTF-8 a quattro byte, ma il tipo di dati CHAR accetta solo caratteri ASCII a byte singolo. Non è possibile caricare caratteri a cinque byte o più lunghi nelle tabelle Amazon Redshift. Per ulteriori informazioni, consulta Caratteri multibyte.

Caricamento di una colonna definita come tipo dati GEOMETRY o GEOGRAPHY

È possibile eseguire COPY su colonne `GEOMETRY` o `GEOGRAPHY` dai dati in un file di testo delimitato da caratteri, ad esempio un file CSV. I dati devono essere nella forma esadecimale del noto formato binario (WKB o EWKB) o del noto formato di testo (WKT o EWKT) e rientrare nella dimensione massima di una singola riga di ingresso al comando COPY. Per ulteriori informazioni, consulta COPY.

Per informazioni su come caricare da uno shapefile, consulta.Caricamento di uno shapefile in Amazon Redshift.

Per ulteriori informazioni sui tipi di dati `GEOMETRY` o `GEOGRAPHY`, consultare Query su dati spaziali in Amazon Redshift.

Caricamento del tipo di dati HLLSKETCH

È possibile copiare gli schizzi HLL solo nel formato sparso o denso supportato da Amazon Redshift. Per utilizzare il comando COPY sugli HyperLogLog schizzi, utilizzate il formato Base64 per schizzi densi e il formato JSON per HyperLogLog schizzi sparsi. HyperLogLog Per ulteriori informazioni, consulta HyperLogLog funzioni.

Nell'esempio seguente vengono importati dati da un file CSV in una tabella utilizzando CREATE TABLE e COPY. Innanzitutto, l'esempio crea la tabella `t1` utilizzando CREATE TABLE.

```
CREATE TABLE t1 (sketch hllsketch, a bigint);
```

Quindi utilizza COPY per importare i dati da un file CSV nella tabella `t1`.

```
COPY t1 FROM s3://DOC-EXAMPLE-BUCKET/unload/' IAM_ROLE
 'arn:aws:iam::0123456789012:role/MyRedshiftRole' NULL AS 'null' CSV;
```

Caricamento di una colonna definita come tipo dati VARBYTE

È possibile caricare dati da un file in formato CSV, Parquet e ORC. Se si sceglie il formato CSV, i dati vengono caricati da un file nella rappresentazione esadecimale dei dati VARBYTE. Non è possibile caricare i dati VARBYTE con l'opzione `FIXEDWIDTH`. Le opzioni `ADDQUOTES` o `REMOVEQUOTES` di COPY non sono supportate. Una colonna VARBYTE non può essere utilizzata come colonna di partizione.

Errori durante la lettura di più file

Il comando COPY è atomico e transazionale. In altre parole, anche quando il comando COPY legge i dati da più file, l'intero processo viene trattato come una singola transazione. Se COPY rileva un errore nella lettura di un file, riprova automaticamente fino a quando il processo scade (vedere statement_timeout) o se i dati non possono essere scaricati da Amazon S3 per un periodo di tempo prolungato (tra 15 e 30 minuti), assicurandosi che ogni file venga caricato una sola volta. Se il comando COPY fallisce, l'intera transazione viene interrotta e tutte le modifiche vengono annullate. Per ulteriori informazioni sulla gestione degli errori di caricamento, vedi Risoluzione di problemi di caricamento dei dati.

Una volta avviato un comando COPY con successo, non fallisce se la sessione termina, ad esempio quando il client si disconnette. Tuttavia, se il comando COPY si trova all'interno di un blocco di transazioni BEGIN … END che non viene completato perché la sessione termina, l'intera transazione, compreso il comando COPY, viene annullata. Per ulteriori informazioni sulle transazioni, consultare BEGIN.

COPY dal formato JSON

La struttura dei dati JSON è costituita da una serie di oggetti o array. Un oggetto JSON inizia e termina con delle parentesi e contiene un insieme non ordinato di coppie nome-valore. Ogni nome e valore sono separati da due punti e le coppie sono separate da virgole. Il nome è una stringa tra doppie virgolette. I caratteri virgoletta devono essere virgolette semplici (0x22), non oblique o "smart".

Un array JSON inizia e termina con delle parentesi e contiene un insieme ordinato di valori separati da virgole. Un valore può essere una stringa tra doppie virgolette, un numero, un booleano vero o falso, nullo, un oggetto JSON o un array.

Gli oggetti e gli array JSON possono essere annidati, consentendo una struttura gerarchica dei dati. L'esempio seguente mostra una struttura di dati JSON con due oggetti validi.

```
{
    "id": 1006410,
    "title": "Amazon Redshift Database Developer Guide"
}
{
    "id": 100540,
    "name": "Amazon Simple Storage Service User Guide"
}
```

Di seguito sono riportati gli stessi dati di due array JSON.

```
[
    1006410,
    "Amazon Redshift Database Developer Guide"
]
[
    100540,
    "Amazon Simple Storage Service User Guide"
]
```

Opzioni COPY per JSON

È possibile specificare le seguenti opzioni quando si utilizza COPY con i dati in formato JSON:

- `'auto'` : COPY carica automaticamente i campi dal file JSON.
- `'auto ignorecase'`: COPY carica automaticamente i campi dal file JSON ignorando la distinzione tra maiuscole e minuscole per i nomi dei campi.
- `s3://jsonpaths_file`: COPY utilizza un file JSONPaths file per analizzare i dati di origine JSON. Un file JSONPath è un file di testo che contiene un singolo oggetto JSON con il nome `"jsonpaths"` accoppiato con un array di espressioni JSONPath. Se il nome è una stringa diversa da `"jsonpaths"`, COPY utilizza l'argomento `'auto'` invece di utilizzare il file JSONPath.

Per esempi che mostrano come caricare i dati usando `'auto'`, `'auto ignorecase'` o un file JSONPaths e usando sia oggetti JSON che array, consultare Esempi di copia da JSON.

Opzione JSONPath

Nella sintassi di COPY di Amazon Redshift, un'espressione JSONPath specifica il percorso esplicito di un singolo elemento del nome in una struttura di dati gerarchica JSON utilizzando la notazione con parentesi o punti. Amazon Redshift non supporta elementi JSONPath, come caratteri jolly o espressioni filtro, che potrebbero determinare un percorso ambiguo o elementi con più nomi. Di conseguenza, Amazon Redshift non è in grado di analizzare strutture di dati complesse e a più livelli.

Di seguito è riportato un esempio di file JSONPath con espressioni JSONPath che utilizzano la notazione a parentesi. Il simbolo del dollaro ($) rappresenta la struttura a livello di root.

```
{
    "jsonpaths": [
       "$['id']",
       "$['store']['book']['title']",
 "$['location'][0]"
    ]
}
```

Nell'esempio precedente, `$['location'][0]` fa riferimento al primo elemento di un array. JSON utilizza un'indicizzazione dell'array basata su zero. Gli indici degli array devono essere numeri interi positivi (maggiori o uguali a zero).

L'esempio seguente mostra il file JSONPath precedente utilizzando la notazione a punti.

```
{
    "jsonpaths": [
       "$.id",
       "$.store.book.title",
 "$.location[0]"
    ]
}
```

Non è possibile mischiare la notazione a parentesi e a punti nell'array `jsonpaths`. Le parentesi possono essere utilizzate sia nella notazione a parentesi sia nella notazione a punti per fare riferimento a un elemento dell'array.

Quando si utilizza la notazione a punti, le espressioni JSONPath non possono contenere i seguenti caratteri:

- Virgolette singole diritte (')
- Periodo o punto (.)
- Parentesi ([ ]) salvo se utilizzate per fare riferimento a un elemento dell'array

Se il valore nella coppia nome-valore a cui fa riferimento un'espressione JSONPath è un oggetto o un array, l'intero oggetto o array viene caricato come stringa, incluse le parentesi tonde o quadre. Ad esempio, si supponga che i dati JSON contengano il seguente oggetto.

```
{
    "id": 0,
    "guid": "84512477-fa49-456b-b407-581d0d851c3c",
    "isActive": true,
    "tags": [
        "nisi",
        "culpa",
        "ad",
        "amet",
        "voluptate",
        "reprehenderit",
        "veniam"
    ],
    "friends": [
        {
            "id": 0,
            "name": "Martha Rivera"
        },
        {
            "id": 1,
            "name": "Renaldo"
        }
    ]
}
```

L'espressione JSONPath `$['tags']` restituisce quindi il seguente valore.

```
"["nisi","culpa","ad","amet","voluptate","reprehenderit","veniam"]"
```

L'espressione JSONPath `$['friends'][1]` restituisce quindi il seguente valore.

```
"{"id": 1,"name": "Renaldo"}"
```

Ogni espressione JSONPath nell'array `jsonpaths` corrisponde a una colonna nella tabella di destinazione Amazon Redshift. L'ordine degli elementi dell'array `jsonpaths` deve corrispondere all'ordine delle colonne della tabella di destinazione o, se viene utilizzato un elenco di colonne, di quest'ultimo.

Per esempi che mostrano come caricare i dati usando l'argomento `'auto'` o un file JSONPath e usando sia oggetti JSON sia array, consultare Esempi di copia da JSON.

Per informazioni su come copiare più file JSON, consultare Utilizzo di un manifest per specificare i fili di dati.

Caratteri escape in JSON

COPY carica `\n` come carattere newline e carica `\t` come carattere di tabulazione. Per caricare una barra rovesciata, crea una sequenza di escape con una barra rovesciata (`\\`).

Ad esempio, supponiamo di avere la seguente struttura JSON in un file denominato `escape.json` nel bucket `s3://mybucket/json/`.

```
{
  "backslash": "This is a backslash: \\",
  "newline": "This sentence\n is on two lines.",
  "tab": "This sentence \t contains a tab."
}
```

Esegui i seguenti comandi per creare la tabella ESCAPES e caricare il JSON.

```
create table escapes (backslash varchar(25), newline varchar(35), tab varchar(35));

copy escapes from 's3://mybucket/json/escape.json'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
format as json 'auto';
```

Eseguire una query sulla tabella ESCAPES per visualizzare i risultati.

```
select * from escapes;
```

```
       backslash        |      newline      |               tab
------------------------+-------------------+----------------------------------
 This is a backslash: \ | This sentence     | This sentence    contains a tab.
                        :  is on two lines.
(1 row)
```

Perdita di precisione numerica

Potrebbe verificarsi una perdita di precisione quando carichi numeri da file di dati in formato JSON in una colonna definita come tipo di dati numerico. Alcuni valori float non sono rappresentati con esattezza nei sistemi informatici. Di conseguenza, i dati copiati da un file JSON potrebbero non essere arrotondati come previsto. Per evitare una perdita di precisione, consigliamo di utilizzare una delle seguenti alternative:

- Rappresentare il numero come stringa racchiudendo il valore in caratteri di doppia virgoletta.
- Utilizza ROUNDEC per arrotondare il numero invece di troncarlo.
- Invece di utilizzare file JSON o Avro, utilizza file di testo CSV, delimitati da caratteri o a larghezza fissa.

COPY da formati di dati a colonna

COPY può caricare i dati da Amazon S3 nei seguenti formati di colonna:

- ORC
- Parquet

Per esempi di utilizzo di COPY con formati di dati colonnari, consulta Esempi di COPY.

COPY supporta dati in formato colonnare con le seguenti considerazioni:

- Il bucket Amazon S3 deve trovarsi nella stessa AWS regione del database Amazon Redshift.
- Per accedere ai dati Amazon S3 tramite un endpoint VPC, configurare l'accesso utilizzando le policy e i ruoli IAM come descritto in Utilizzo di Amazon Redshift Spectrum con il routing VPC avanzato nella Guida alla gestione di Amazon Redshift.
- COPY non applica automaticamente le codifiche di compressione.
- Sono supportati solo i seguenti parametri COPY:

  - ACCEPTINVCHARS durante la copia da un file ORC o Parquet.
  - FILLRECORD
  - FROM
  - IAM_ROLE
  - CREDENTIALS
  - STATUPDATE
  - MANIFEST
  - EXPLICIT_IDS
- Se COPY rileva un errore durante il caricamento, il comando fallisce. ACCEPTANYDATE e MAXERROR non sono supportati per i tipi di dati colonnari..
- I messaggi di errore vengono inviati al client SQL. Alcuni errori vengono registrati in STL_LOAD_ERRORS e STL_ERROR.
- COPY inserisce i valori nelle colonne della tabella di destinazione nello stesso ordine in cui si presentano le colonne nei file di dati a colonna. Il numero di colonne nella tabella di destinazione e il numero di colonne nel file di dati devono corrispondere.
- Se il file specificato per l'operazione COPY include una delle seguenti estensioni, decomprimiamo i dati senza la necessità di aggiungere alcun parametro:
  - `.gz`
  - `.snappy`
  - `.bz2`
- Il COPY dai formati di file Parquet e ORC utilizza Redshift Spectrum e l'accesso al bucket. Per utilizzare COPY per questi formati, assicurati che non vi siano policy IAM che blocchino l'uso di URL predefiniti di Amazon S3. Gli URL predefiniti generati da Amazon Redshift sono validi per 1 ora, in modo che Amazon Redshift abbia abbastanza tempo per caricare tutti i file dal bucket Amazon S3. Viene generato un URL predefinito univoco per ogni file scansionato da COPY da formati di dati colonnari. Per le policy bucket che includono un'`s3:signatureAge`azione, assicurati di impostare il valore su almeno 3.600.000 millisecondi. Per ulteriori informazioni, consultare Utilizzo di Amazon Redshift Spectrum con il routing VPC avanzato.

Stringhe DATEFORMAT e TIMEFORMAT

Il comando COPY utilizza le opzioni DATEFORMAT e TIMEFORMAT per analizzare i valori di data e ora nei dati di origine. DATEFORMAT e TIMEFORMAT sono stringhe formattate che devono

corrispondere al formato dei valori di data e ora dei dati di origine. Ad esempio, un comando COPY che carica i dati di origine con il valore della data `Jan-01-1999` deve includere la seguente stringa DATEFORMAT:

```
COPY ...
            DATEFORMAT AS 'MON-DD-YYYY'
```

Per ulteriori informazioni sulla gestione delle conversioni dei dati COPY, consulta Parametri di conversione dei dati.

Le stringhe DATEFORMAT e TIMEFORMAT possono contenere separatori datetime (come "`-`", "`/`" o "`:`") e i formati datepart e timepart nella tabella seguente.

> **Note**
>
> Se non è possibile abbinare il formato dei valori di data e ora con i seguenti datepart e timepart o se i valori di data e ora utilizzano formati diversi tra di loro, utilizza l'argomento `'auto'` con il parametro DATEFORMAT o TIMEFORMAT. L'argomento `'auto'` riconosce diversi formati che non sono supportati quando si utilizza una stringa DATEFORMAT o TIMEFORMAT. Per ulteriori informazioni, consulta Utilizzo del riconoscimento automatico con DATEFORMAT e TIMEFORMAT.

Parte di data o parte di ora	Significato
YY	Anno senza secolo
YYYY	Anno con secolo
MM	Mese espresso come numero
MON	Mese come nome (abbreviato o completo)
DD	Giorno del mese espresso come numero
HH o HH24	Ora (orologio da 24 ore)

Parte di data o parte di ora	Significato
	Note Nel formato delle stringhe DATETIME per le funzioni SQL, HH è lo stesso di HH12. Tuttavia, nelle stringhe DATEFORMAT e TIMEFORMAT per COPY, HH è uguale a HH24.
HH12	Ora (orologio da 12 ore)
MI	Minuti
SS	Secondi
AM o PM	Indicatore meridiano (per orologio a 12 ore)

Il formato predefinito per la data è YYYY-MM-DD. Il timestamp predefinito senza fuso orario (TIMESTAMP) è AAAA-MM-GG HH:MI:SS. Il formato di timestamp di default con fuso orario (TIMESTAMPTZ) è AAAA-MM-GD H:MI:SSOF, dove OF è l'offset rispetto a UTC (ad esempio, -8:00). Non è possibile includere un specificatore di fuso orario (TZ, tz o OF) nel timeformat_string. Il campo dei secondi (SS) supporta anche i secondi frazionari fino a un livello di dettaglio in microsecondi. Per caricare i dati TIMESTAMPTZ in un formato diverso da quello predefinito, specificare "auto".

Di seguito sono riportati alcuni esempi di date o ore che è possibile trovare nei dati di origine e le relative stringhe DATEFORMAT o TIMEFORMAT.

Esempio di valori di data o ora nei dei dati di origine	Sintassi di DATEFORMAT o TIMEFORMAT
03/31/2003	DATEFORMAT AS 'MM/DD/YYYY'
31 marzo 2003	DATEFORMAT AS 'MON DD, YYYY'

Esempio di valori di data o ora nei dei dati di origine	Sintassi di DATEFORMAT o TIMEFORMAT
03.31.2003 18:45:05 03.31.2003 18:45:05.123456	TIMEFORMAT AS 'MM.DD.YYYY HH:MI:SS'

Esempio

Per un esempio di utilizzo di TIMEFORMAT, consulta Caricamento di un Timestamp o di un Datestamp.

Utilizzo del riconoscimento automatico con DATEFORMAT e TIMEFORMAT

Se si specifica `'auto'` come argomento per il parametro DATEFORMAT o TIMEFORMAT, Amazon Redshift riconoscerà e convertirà automaticamente il formato data o ora nei dati di origine. Di seguito viene riportato un esempio.

```
copy favoritemovies from 'dynamodb://ProductCatalog'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
dateformat 'auto';
```

Se usato con l'argomento `'auto'` per DATEFORMAT e TIMEFORMAT, COPY riconosce e converte i formati di data e ora elencati nella tabella in Stringhe DATEFORMAT e TIMEFORMAT. Inoltre, l'argomento `'auto'` riconosce i seguenti formati che non sono supportati quando si usa una stringa DATEFORMAT e TIMEFORMAT.

Formato	Esempio di stringa di ingresso valida
ISO 8601	2019-02-11T05:09:12.195Z
Giuliano	J2451187
BC	Gen-08-95 BC
YYYYMMDD HHMISS	19960108 040809
YYMMDD HHMISS	960108 040809

Formato	Esempio di stringa di ingresso valida
YYYY.DDD	1996.008
YYYY-MM-DD HH:MI:SS.SSS	1996-01-08 04:05:06.789
DD Mese HH:MI:SS YYYY TZ	17 Dic 07:37:16 1997 PST
MM/DD/YYYY HH:MI:SS.SS TZ	12/17/1997 07:37:16.00 PST
YYYY-MM-DD HH:MI:SS+/-TZ	1997-12-17 07:37:16-08
DD.MM.YYYY HH:MI:SS TZ	12.17.1997 07:37:16.00 PST

Il riconoscimento automatico non supporta epochsecs ed epochmillisecs.

Per verificare se un valore di data o di timestamp viene convertito automaticamente, utilizza una funzione CAST per tentare di convertire la stringa in un valore di data o timestamp. Ad esempio, i seguenti comandi verificano il valore del timestamp `'J2345678 04:05:06.789'`:

```
create table formattest (test char(21));
insert into formattest values('J2345678 04:05:06.789');
select test, cast(test as timestamp) as timestamp, cast(test as date) as date from
 formattest;

        test          |      timestamp      | date
----------------------+---------------------+------------
J2345678 04:05:06.789   1710-02-23 04:05:06 1710-02-23
```

Se i dati sorgente di una colonna DATE includono informazioni sull'ora, la componente temporale viene troncata. Se i dati sorgente di una colonna TIMESTAMP omettono informazioni sull'ora, per la componente dell'ora viene utilizzato 00:00:00.

Esempi di COPY

> **Note**
>
> Questi esempi contengono interruzioni di riga per una migliore leggibilità. Non includere interruzioni di riga o spazi nella stringa credentials-args.

Argomenti

Caricamento di FAVORITEMOVIES da una tabella DynamoDB

Gli AWS SDK includono un semplice esempio di creazione di una tabella DynamoDB chiamata Movies. (Per questo esempio, consultare Nozioni di base su DynamoDB.) Nell'esempio seguente la tabella MOVIES di Amazon Redshift viene caricata con i dati della tabella DynamoDB. La tabella di Amazon Redshift deve esistere già nel database.

```
copy favoritemovies from 'dynamodb://Movies'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
readratio 50;
```

Caricamento di LISTING da un bucket Amazon S3

L'esempio seguente carica LISTING da un bucket Amazon S3. Il comando COPY carica tutti i file contenuti nella cartella `/data/listing/`.

```
copy listing
from 's3://mybucket/data/listing/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Caricamento di LISTING da un cluster Amazon EMR

L'esempio seguente carica la tabella SALES con i dati delimitati da schede da file compressi lzop in un cluster Amazon EMR. COPY carica ogni file nella cartella `myoutput/` che inizia con `part-`.

```
copy sales
from 'emr://j-SAMPLE2B500FC/myoutput/part-*'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '\t' lzop;
```

L'esempio seguente carica la tabella SALES con i dati con dati formattati in JSON in un cluster Amazon EMR. COPY carica ogni file nella cartella `myoutput/json/`.

```
copy sales
from 'emr://j-SAMPLE2B500FC/myoutput/json/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
JSON 's3://mybucket/jsonpaths.txt';
```

Utilizzo di un manifest per specificare i fili di dati

È possibile utilizzare un manifest per assicurarsi che il comando COPY carichi tutti i file richiesti, e solo i file richiesti, da Amazon S3. È anche possibile utilizzare un manifest quando è necessario caricare più file da bucket o file diversi che non condividono lo stesso prefisso.

Ad esempio, supponi di dover caricare i seguenti tre file: `custdata1.txt`, `custdata2.txt` e `custdata3.txt`. È possibile usare il seguente comando per caricare tutti i file in `mybucket` che iniziano con `custdata` specificando un prefisso:

```
copy category
from 's3://mybucket/custdata'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Se solo due dei file esistono a causa di un errore, COPY carica solo questi due file e terminerà con successo, con un caricamento di dati che risulterà incompleto. Se il bucket contiene anche un file indesiderato che utilizza lo stesso prefisso, come ad esempio un file chiamato `custdata.backup`, COPY carica anche quel file, caricando così dati indesiderati.

Per assicurarsi che tutti i file richiesti siano caricati e per evitare che vengano caricati file indesiderati, è possibile utilizzare un file manifest. Il manifest è un file di testo con formattazione JSON che elenca i file che devono essere elaborati dal comando COPY. Ad esempio, il seguente manifest carica i tre file nell'esempio precedente.

```
{
   "entries":[
      {
         "url":"s3://mybucket/custdata.1",
         "mandatory":true
      },
      {
         "url":"s3://mybucket/custdata.2",
         "mandatory":true
      },
      {
         "url":"s3://mybucket/custdata.3",
         "mandatory":true
      }
   ]
}
```

Il flag opzionale `mandatory` indica se COPY deve terminare se il file non esiste. Il valore predefinito è `false`. Indipendentemente da eventuali impostazioni obbligatorie, COPY termina se non vengono trovati file. In questo esempio, COPY restituisce un errore se uno qualsiasi dei file non viene trovato. I file indesiderati che potrebbero essere stati raccolti se hai specificato solo un prefisso della chiave, come `custdata.backup`, vengono ignorati, perché non sono sul manifest.

Quando si carica da file di dati in formato ORC o Parquet, è necessario un campo `meta`, come mostrato nell'esempio seguente.

```
{
   "entries":[
      {
         "url":"s3://mybucket-alpha/orc/2013-10-04-custdata",
         "mandatory":true,
         "meta":{
            "content_length":99
         }
      },
      {
         "url":"s3://mybucket-beta/orc/2013-10-05-custdata",
         "mandatory":true,
         "meta":{
            "content_length":99
         }
      }
   ]
}
```

L'esempio seguente utilizza un manifest denominato `cust.manifest`.

```
copy customer
from 's3://mybucket/cust.manifest'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
format as orc
manifest;
```

È possibile utilizzare un manifest per caricare file da diversi bucket o file che non condividono lo stesso prefisso. L'esempio seguente mostra il JSON per caricare dati con file i cui nomi iniziano con uno stamp di data.

```
{
```

```
  "entries": [
    {"url":"s3://mybucket/2013-10-04-custdata.txt","mandatory":true},
    {"url":"s3://mybucket/2013-10-05-custdata.txt","mandatory":true},
    {"url":"s3://mybucket/2013-10-06-custdata.txt","mandatory":true},
    {"url":"s3://mybucket/2013-10-07-custdata.txt","mandatory":true}
  ]
}
```

Il manifest può elencare i file che si trovano in bucket diversi, purché i bucket si trovino nella stessa AWS regione del cluster.

```
{
  "entries": [
    {"url":"s3://mybucket-alpha/custdata1.txt","mandatory":false},
    {"url":"s3://mybucket-beta/custdata1.txt","mandatory":false},
    {"url":"s3://mybucket-beta/custdata2.txt","mandatory":false}
  ]
}
```

Caricamento di LISTING da un file delimitato da pipe (Delimitatore predefinito)

L'esempio seguente è un caso molto semplice in cui non sono specificate opzioni e il file di input contiene il delimitatore predefinito, un carattere pipe ("|").

```
copy listing
from 's3://mybucket/data/listings_pipe.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Caricamento di LISTING utilizzando i dati a colonna nel formato Parquet

Il seguente esempio carica i dati da una cartella su Amazon S3 chiamata parquet.

```
copy listing
from 's3://mybucket/data/listings/parquet/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
format as parquet;
```

Caricamento di LISTING utilizzando i dati a colonna nel formato ORC

Il seguente esempio carica i dati da una cartella su Amazon S3 chiamata `orc`.

```
copy listing
from 's3://mybucket/data/listings/orc/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
format as orc;
```

Caricamento di EVENT con le opzioni

Il seguente esempio carica i dati delimitati da pipe nella tabella EVENT e applica le seguenti regole:

- Se si utilizzano coppie di virgolette per circondare una qualsiasi stringa di caratteri, vengono rimosse.
- Sia le stringhe vuote sia quelle che contengono spazi vuoti vengono caricate come valori NULL.
- Il caricamento non riesce se vengono restituiti più di 5 errori.
- I valori di timestamp devono essere conformi al formato specificato; ad esempio, un timestamp valido è `2008-09-26 05:43:12`.

```
copy event
from 's3://mybucket/data/allevents_pipe.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
removequotes
emptyasnull
blanksasnull
maxerror 5
delimiter '|'
timeformat 'YYYY-MM-DD HH:MI:SS';
```

Caricamento di VENUE da un file di dati a larghezza fissa

```
copy venue
from 's3://mybucket/data/venue_fw.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
fixedwidth 'venueid:3,venuename:25,venuecity:12,venuestate:2,venueseats:6';
```

L'esempio precedente assume un file di dati formattato nello stesso modo dei dati campione mostrati. Nel seguente esempio, gli spazi fungono da placeholder in modo che tutte le colonne abbiano la stessa larghezza indicata nelle specifiche:

```
1  Toyota Park              Bridgeview  IL0
```

```
2  Columbus Crew Stadium    Columbus    OH0
3  RFK Stadium              Washington  DC0
4  CommunityAmerica BallparkKansas City KS0
5  Gillette Stadium         Foxborough  MA68756
```

Caricamento di CATEGORY da un file CSV

Supponi di voler caricare CATEGORY con i valori indicati nella tabella seguente.

catid	catgroup	catname	catdesc
12	Spettacoli	Musical	Musical teatrali
13	Spettacoli	Rappresentazioni	Tutto il teatro non musicale
14	Spettacoli	Opera	Tutta l'opera lirica, leggera e "rock"
15	Concerti	Classici	Tutti i concerti sinfonici, concertistici e corali

L'esempio seguente mostra il contenuto di un file di testo con i valori dei campi separati da virgole.

```
12,Shows,Musicals,Musical theatre
13,Shows,Plays,All "non-musical" theatre
14,Shows,Opera,All opera, light, and "rock" opera
15,Concerts,Classical,All symphony, concerto, and choir concerts
```

Se carichi il file utilizzando il parametro DELIMITER per specificare l'input delimitato da virgole, il comando COPY non riesce perché alcuni campi di input contengono virgole. È possibile evitare questo problema utilizzando il parametro CSV e racchiudendo i campi che contengono virgole tra virgolette. Se il carattere virgolette viene visualizzato all'interno di una stringa tra virgolette, è necessario eseguirne l'escape raddoppiando le virgolette. Il carattere di virgoletta di default è una virgoletta doppia, quindi è necessario eseguire l'escape di ogni virgoletta doppia con una virgoletta doppia aggiuntiva. Il nuovo file di input è simile a questo.

```
12,Shows,Musicals,Musical theatre
13,Shows,Plays,"All ""non-musical"" theatre"
14,Shows,Opera,"All opera, light, and ""rock"" opera"
```

```
15,Concerts,Classical,"All symphony, concerto, and choir concerts"
```

Supponendo che il nome del file sia `category_csv.txt`, è possibile caricare il file utilizzando il seguente comando COPY:

```
copy category
from 's3://mybucket/data/category_csv.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
csv;
```

In alternativa, per evitare di dover creare una sequenza di escape per le doppie virgolette nell'input, è possibile specificare un carattere virgolette diverso utilizzando il parametro QUOTE AS. Per esempio, la seguente versione di `category_csv.txt` utilizza "%" come carattere virgolette.

```
12,Shows,Musicals,Musical theatre
13,Shows,Plays,%All "non-musical" theatre%
14,Shows,Opera,%All opera, light, and "rock" opera%
15,Concerts,Classical,%All symphony, concerto, and choir concerts%
```

Il seguente comando COPY utilizza QUOTE AS per caricare `category_csv.txt`:

```
copy category
from 's3://mybucket/data/category_csv.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
csv quote as '%';
```

Caricamento di VENUE con valori espliciti per una colonna IDENTITY

L'esempio seguente presuppone che quando è stata creata la tabella VENUE almeno una colonna (come la colonna `venueid`) sia stata specificata come colonna IDENTITY. Questo comando sostituisce il comportamento di IDENTITY predefinito dei valori autogeneranti per una colonna IDENTITY e carica invece i valori espliciti dal file venue.txt. Amazon Redshift non verifica se i valori IDENTITY duplicati vengono caricati nella tabella quando utilizza l'opzione EXLICIT_IDS.

```
copy venue
from 's3://mybucket/data/venue.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
explicit_ids;
```

Caricamento di TIME da un file GZIP delimitato da pipe

L'esempio seguente carica la tabella TIME da un file GZIP delimitato dal pipe:

```
copy time
from 's3://mybucket/data/timerows.gz'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
gzip
delimiter '|';
```

Caricamento di un Timestamp o di un Datestamp

L'esempio seguente carica i dati con un timestamp formattato.

> **Note**
>
> Il TIMEFORMAT di `HH:MI:SS` può anche supportare i secondi frazionari oltre il `SS` a un livello di dettaglio di microsecondi. Il file `time.txt` utilizzato in questo esempio contiene una riga `2009-01-12 14:15:57.119568`.

```
copy timestamp1
from 's3://mybucket/data/time.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
timeformat 'YYYY-MM-DD HH:MI:SS';
```

Il risultato di questa copia è il seguente:

```
select * from timestamp1;
c1
----------------------------
2009-01-12 14:15:57.119568
(1 row)
```

Caricamento dei dati da un file con valori predefiniti

L'esempio seguente usa una variazione della tabella VENUE nel database TICKIT. Considera una tabella VENUE_NEW definita con la seguente dichiarazione:

```
create table venue_new(
venueid smallint not null,
```

```
venuename varchar(100) not null,
venuecity varchar(30),
venuestate char(2),
venueseats integer not null default '1000');
```

Considera un file di dati venue_noseats.txt che non contiene valori per la colonna VENUESEATS, come mostrato nell'esempio seguente:

```
1|Toyota Park|Bridgeview|IL|
2|Columbus Crew Stadium|Columbus|OH|
3|RFK Stadium|Washington|DC|
4|CommunityAmerica Ballpark|Kansas City|KS|
5|Gillette Stadium|Foxborough|MA|
6|New York Giants Stadium|East Rutherford|NJ|
7|BMO Field|Toronto|ON|
8|The Home Depot Center|Carson|CA|
9|Dick's Sporting Goods Park|Commerce City|CO|
10|Pizza Hut Park|Frisco|TX|
```

La seguente dichiarazione COPY caricherà la tabella dal file e applicherà il valore DEFAULT ("1000") alla colonna omessa:

```
copy venue_new(venueid, venuename, venuecity, venuestate)
from 's3://mybucket/data/venue_noseats.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|';
```

Ora visualizza la tabella caricata:

```
select * from venue_new order by venueid;
venueid |         venuename          |    venuecity    | venuestate | venueseats
---------+----------------------------+-----------------+------------+------------
1 | Toyota Park                | Bridgeview      | IL         |       1000
2 | Columbus Crew Stadium      | Columbus        | OH         |       1000
3 | RFK Stadium                | Washington      | DC         |       1000
4 | CommunityAmerica Ballpark  | Kansas City     | KS         |       1000
5 | Gillette Stadium           | Foxborough      | MA         |       1000
6 | New York Giants Stadium    | East Rutherford | NJ         |       1000
7 | BMO Field                  | Toronto         | ON         |       1000
8 | The Home Depot Center      | Carson          | CA         |       1000
9 | Dick's Sporting Goods Park | Commerce City   | CO         |       1000
10 | Pizza Hut Park             | Frisco          | TX         |       1000
```

```
(10 rows)
```

Per l'esempio seguente, oltre a supporre che nel file non siano inclusi dati VENUESEATS, supponi anche che non siano inclusi dati VENUENAME:

```
1||Bridgeview|IL|
2||Columbus|OH|
3||Washington|DC|
4||Kansas City|KS|
5||Foxborough|MA|
6||East Rutherford|NJ|
7||Toronto|ON|
8||Carson|CA|
9||Commerce City|CO|
10||Frisco|TX|
```

Usando la stessa definizione della tabella, la seguente dichiarazione COPY non riesce perché non è stato specificato alcun valore DEFAULT per VENUENAME, il quale è una colonna NOT NULL:

```
copy venue(venueid, venuecity, venuestate)
from 's3://mybucket/data/venue_pipe.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|';
```

Considera ora una variazione della tabella VENUE che utilizza una colonna IDENTITY:

```
create table venue_identity(
venueid int identity(1,1),
venuename varchar(100) not null,
venuecity varchar(30),
venuestate char(2),
venueseats integer not null default '1000');
```

Come per l'esempio precedente, supponi che la colonna VENUESEATS non abbia valori corrispondenti nel file sorgente. La seguente dichiarazione COPY carica correttamente la tabella, inclusi i valori dei dati IDENTITY predefiniti, invece di generare tali valori autonomamente:

```
copy venue(venueid, venuename, venuecity, venuestate)
from 's3://mybucket/data/venue_pipe.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|' explicit_ids;
```

Questa dichiarazione non riesce perché non include la colonna IDENTITY (VENUEID è assente dall'elenco delle colonne) ma include un parametro EXPLICIT_IDS:

```
copy venue(venuename, venuecity, venuestate)
from 's3://mybucket/data/venue_pipe.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|' explicit_ids;
```

Questa dichiarazione non riesce perché non include un parametro EXPLICIT_IDS:

```
copy venue(venueid, venuename, venuecity, venuestate)
from 's3://mybucket/data/venue_pipe.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|';
```

Dati COPY con l'opzione ESCAPE

L'esempio seguente mostra come caricare i caratteri che corrispondono al carattere del delimitatore (in questo caso, il carattere pipe). Nel file di input, assicurati che tutti i caratteri pipe (|) che desideri caricare siano resi una sequenza di escape con il carattere di barra rovesciata (\). Quindi carica il file con il parametro ESCAPE.

```
$ more redshiftinfo.txt
1|public\|event\|dwuser
2|public\|sales\|dwuser

create table redshiftinfo(infoid int,tableinfo varchar(50));

copy redshiftinfo from 's3://mybucket/data/redshiftinfo.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter '|' escape;

select * from redshiftinfo order by 1;
infoid |       tableinfo
-------+--------------------
1      | public|event|dwuser
2      | public|sales|dwuser
(2 rows)
```

Senza il parametro ESCAPE, questo comando COPY fallisce con un errore `Extra column(s) found`.

> **Important**
>
> Se carichi i dati utilizzando un COPY con il parametro ESCAPE, è necessario specificare anche il parametro ESCAPE con il comando UNLOAD per generare il file di output reciproco. Allo stesso modo, se utilizzi UNLOAD usando il parametro ESCAPE, è necessario usare ESCAPE quando effettui un COPY sugli stessi dati.

Esempi di copia da JSON

Negli esempi seguenti, carica la tabella CATEGORY con i seguenti dati.

CATID	CATGROUP	CATNAME	CATDESC
1	Sport	MLB	Major League Baseball
2	Sport	NHL	National Hockey League
3	Sport	NFL	National Football League
4	Sport	NBA	National Basketball Association
5	Concerti	Classici	Tutti i concerti sinfonici, concertistici e corali

Argomenti

- Caricamento da dati JSON utilizzando l'opzione "auto".
- Caricamento da dati JSON utilizzando l'opzione 'auto ignorecase'
- Caricamento da dati JSON utilizzando un file JSONPath
- Caricamento da array JSON utilizzando un file JSONPath

Caricamento da dati JSON utilizzando l'opzione "auto".

Per caricare dai dati JSON utilizzando l'opzione `'auto'`, i dati JSON devono essere costituiti da un set di oggetti. I nomi delle chiavi devono corrispondere ai nomi delle colonne, ma l'ordine non ha importanza. Di seguito viene mostrato il contenuto di un file denominato `category_object_auto.json`.

```
{
    "catdesc": "Major League Baseball",
    "catid": 1,
    "catgroup": "Sports",
    "catname": "MLB"
}
{
    "catgroup": "Sports",
    "catid": 2,
    "catname": "NHL",
    "catdesc": "National Hockey League"
}{
    "catid": 3,
    "catname": "NFL",
    "catgroup": "Sports",
    "catdesc": "National Football League"
}
{
    "bogus": "Bogus Sports LLC",
    "catid": 4,
    "catgroup": "Sports",
    "catname": "NBA",
    "catdesc": "National Basketball Association"
}
{
    "catid": 5,
    "catgroup": "Shows",
    "catname": "Musicals",
    "catdesc": "All symphony, concerto, and choir concerts"
}
```

Per caricare dal file dati JSON nell'esempio precedente, emettere il seguente comando COPY.

```
copy category
from 's3://mybucket/category_object_auto.json'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
json 'auto';
```

Caricamento da dati JSON utilizzando l'opzione 'auto ignorecase'

Per caricare dai dati JSON utilizzando l'opzione `'auto ignorecase'`, i dati JSON devono essere costituiti da un set di oggetti. Il formato maiuscolo/minuscolo dei nomi delle chiavi non deve

corrispondere ai nomi delle colonne e l'ordine non ha importanza. Di seguito viene mostrato il contenuto di un file denominato `category_object_auto-ignorecase.json`.

```
{
    "CatDesc": "Major League Baseball",
    "CatID": 1,
    "CatGroup": "Sports",
    "CatName": "MLB"
}
{
    "CatGroup": "Sports",
    "CatID": 2,
    "CatName": "NHL",
    "CatDesc": "National Hockey League"
}{
    "CatID": 3,
    "CatName": "NFL",
    "CatGroup": "Sports",
    "CatDesc": "National Football League"
}
{
    "bogus": "Bogus Sports LLC",
    "CatID": 4,
    "CatGroup": "Sports",
    "CatName": "NBA",
    "CatDesc": "National Basketball Association"
}
{
    "CatID": 5,
    "CatGroup": "Shows",
    "CatName": "Musicals",
    "CatDesc": "All symphony, concerto, and choir concerts"
}
```

Per caricare dal file di dati JSON nell'esempio precedente, emettere il seguente comando COPY.

```
copy category
from 's3://mybucket/category_object_auto ignorecase.json'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
json 'auto ignorecase';
```

Caricamento da dati JSON utilizzando un file JSONPath

Se gli oggetti dei dati JSON non corrispondono direttamente ai nomi delle colonne, è possibile utilizzare un file JSONPath per mappare gli elementi JSON sulle colonne. Anche in questo caso, l'ordine non ha importanza nei dati di origine JSON, ma l'ordine delle espressioni dei file JSONPath deve corrispondere all'ordine delle colonne. Supponi di avere il seguente file di dati, denominato `category_object_paths.json`.

```
{
    "one": 1,
    "two": "Sports",
    "three": "MLB",
    "four": "Major League Baseball"
}
{
    "three": "NHL",
    "four": "National Hockey League",
    "one": 2,
    "two": "Sports"
}
{
    "two": "Sports",
    "three": "NFL",
    "one": 3,
    "four": "National Football League"
}
{
    "one": 4,
    "two": "Sports",
    "three": "NBA",
    "four": "National Basketball Association"
}
{
    "one": 6,
    "two": "Shows",
    "three": "Musicals",
    "four": "All symphony, concerto, and choir concerts"
}
```

Il seguente file JSONPath, denominato `category_jsonpath.json`, mappa i dati sorgente nelle colonne della tabella.

```
{
    "jsonpaths": [
        "$['one']",
        "$['two']",
        "$['three']",
        "$['four']"
    ]
}
```

Per caricare dal file di dati JSON nell'esempio precedente, emettere il seguente comando COPY.

```
copy category
from 's3://mybucket/category_object_paths.json'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
json 's3://mybucket/category_jsonpath.json';
```

Caricamento da array JSON utilizzando un file JSONPath

Per caricare dai dati JSON costituiti da un insieme di array, è necessario utilizzare un file JSONPath per mappare gli elementi dell'array sulle colonne. Supponi di avere il seguente file di dati, denominato `category_array_data.json`.

```
[1,"Sports","MLB","Major League Baseball"]
[2,"Sports","NHL","National Hockey League"]
[3,"Sports","NFL","National Football League"]
[4,"Sports","NBA","National Basketball Association"]
[5,"Concerts","Classical","All symphony, concerto, and choir concerts"]
```

Il seguente file JSONPath, denominato `category_array_jsonpath.json`, mappa i dati sorgente nelle colonne della tabella.

```
{
    "jsonpaths": [
        "$[0]",
        "$[1]",
        "$[2]",
        "$[3]"
    ]
}
```

Per caricare dal file di dati JSON nell'esempio precedente, emettere il seguente comando COPY.

```
copy category
from 's3://mybucket/category_array_data.json'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
json 's3://mybucket/category_array_jsonpath.json';
```

Esempi di copia da Avro

Negli esempi seguenti, carica la tabella CATEGORY con i seguenti dati.

CATID	CATGROUP	CATNAME	CATDESC
1	Sport	MLB	Major League Baseball
2	Sport	NHL	National Hockey League
3	Sport	NFL	National Football League
4	Sport	NBA	National Basketball Association
5	Concerti	Classici	Tutti i concerti sinfonici, concertistici e corali

Argomenti

- Caricamento dai dati Avro utilizzando l'opzione "auto".
- Caricamento da dati Avro utilizzando l'opzione 'auto ignorecase'
- Caricamento dai dati Avro utilizzando un file JSONPath

Caricamento dai dati Avro utilizzando l'opzione "auto".

Per caricare dai dati Avro usando l'argomento `'auto'`, i nomi dei campi nello schema Avro devono corrispondere ai nomi delle colonne. Quando si utilizza l'argomento `'auto'`, l'ordine non ha importanza. Di seguito viene mostrato lo schema per un file denominato `category_auto.avro`.

```
{
    "name": "category",
    "type": "record",
    "fields": [
        {"name": "catid", "type": "int"},
```

```
        {"name": "catdesc", "type": "string"},
        {"name": "catname", "type": "string"},
        {"name": "catgroup", "type": "string"},
}
```

I dati in un file Avro sono in formato binario, quindi non sono leggibili dall'utente. Di seguito viene mostrata una rappresentazione JSON dei dati nel file `category_auto.avro`.

```
{
   "catid": 1,
   "catdesc": "Major League Baseball",
   "catname": "MLB",
   "catgroup": "Sports"
}
{
   "catid": 2,
   "catdesc": "National Hockey League",
   "catname": "NHL",
   "catgroup": "Sports"
}
{
   "catid": 3,
   "catdesc": "National Basketball Association",
   "catname": "NBA",
   "catgroup": "Sports"
}
{
   "catid": 4,
   "catdesc": "All symphony, concerto, and choir concerts",
   "catname": "Classical",
   "catgroup": "Concerts"
}
```

Per caricare dal file di dati Avro nell'esempio precedente, emettere il seguente comando COPY.

```
copy category
from 's3://mybucket/category_auto.avro'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
format as avro 'auto';
```

Caricamento da dati Avro utilizzando l'opzione 'auto ignorecase'

Per caricare dai dati Avro con l'argomento `'auto ignorecase'`, il formato maiuscolo/minuscolo dei nomi dei campi nello schema Avro non deve corrispondere al formato dei nomi delle colonne. Quando si utilizza l'argomento `'auto ignorecase'`, l'ordine non ha importanza. Di seguito viene mostrato lo schema per un file denominato `category_auto-ignorecase.avro`.

```
{
    "name": "category",
    "type": "record",
    "fields": [
        {"name": "CatID", "type": "int"},
        {"name": "CatDesc", "type": "string"},
        {"name": "CatName", "type": "string"},
        {"name": "CatGroup", "type": "string"},
}
```

I dati in un file Avro sono in formato binario, quindi non sono leggibili dall'utente. Di seguito viene mostrata una rappresentazione JSON dei dati nel file `category_auto-ignorecase.avro`.

```
{
   "CatID": 1,
   "CatDesc": "Major League Baseball",
   "CatName": "MLB",
   "CatGroup": "Sports"
}
{
   "CatID": 2,
   "CatDesc": "National Hockey League",
   "CatName": "NHL",
   "CatGroup": "Sports"
}
{
   "CatID": 3,
   "CatDesc": "National Basketball Association",
   "CatName": "NBA",
   "CatGroup": "Sports"
}
{
   "CatID": 4,
   "CatDesc": "All symphony, concerto, and choir concerts",
   "CatName": "Classical",
```

```
   "CatGroup": "Concerts"
}
```

Per caricare dal file di dati Avro nell'esempio precedente, emettere il seguente comando COPY.

```
copy category
from 's3://mybucket/category_auto-ignorecase.avro'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
format as avro 'auto ignorecase';
```

Caricamento dai dati Avro utilizzando un file JSONPath

Se i nomi dei campi nello schema Avro non corrispondono direttamente ai nomi delle colonne, è possibile utilizzare un file JSONPath per mappare gli elementi dello schema sulle colonne. L'ordine delle espressioni dei file JSONPath deve corrispondere all'ordine delle colonne.

Supponi di avere un file di dati chiamato `category_paths.avro` che contenga gli stessi dati dell'esempio precedente, ma con lo schema seguente.

```
{
    "name": "category",
    "type": "record",
    "fields": [
        {"name": "id", "type": "int"},
        {"name": "desc", "type": "string"},
        {"name": "name", "type": "string"},
        {"name": "group", "type": "string"},
        {"name": "region", "type": "string"}
     ]
}
```

Il seguente file JSONPath, denominato `category_path.avropath`, mappa i dati sorgente nelle colonne della tabella.

```
{
    "jsonpaths": [
        "$['id']",
        "$['group']",
        "$['name']",
        "$['desc']"
    ]
```

```
}
```

Per caricare dal file di dati Avro nell'esempio precedente, emettere il seguente comando COPY.

```
copy category
from 's3://mybucket/category_object_paths.avro'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
format avro 's3://mybucket/category_path.avropath ';
```

Preparazione dei file per COPY con l'opzione ESCAPE

L'esempio seguente descrive come preparare i dati per eseguire un "escape" dei caratteri newline prima di importarli in una tabella Amazon Redshift usando il comando COPY con il parametro ESCAPE. Senza preparare i dati per delimitare i caratteri newline, Amazon Redshift restituisce errori di caricamento quando si esegue il comando COPY, perché il carattere newline è normalmente usato come separatore di record.

Ad esempio, si consideri un file o una colonna in una tabella esterna che si desidera copiare in una tabella Amazon Redshift. Se il file o la colonna ha del contenuto formattato in XML o dati simili, è necessario assicurarsi che tutti i caratteri newline (\n) che fanno parte del contenuto siano inseriti in una sequenza di escape con il carattere di barra rovesciata (\).

Un file o una tabella che contiene caratteri newline incorporati è che fornisce un modello relativamente facile da associare. Ogni carattere newline incorporato molto probabilmente segue sempre un carattere > con potenzialmente alcuni caratteri di spazio (' ' o tabulazione) in mezzo, come è possibile vedere nel seguente esempio di un file di testo denominato `nlTest1.txt`.

```
$ cat nlTest1.txt
<xml start>
<newline characters provide>
<line breaks at the end of each>
<line in content>
</xml>|1000
<xml>
</xml>|2000
```

Con l'esempio seguente, è possibile eseguire un'utilità di elaborazione del testo per pre-elaborare il file sorgente e inserire caratteri di escape, se necessario. (Il carattere | deve essere usato come delimitatore per separare i dati delle colonne quando viene copiato in una tabella Amazon Redshift).

```
$ sed -e ':a;N;$!ba;s/>[[:space:]]*\n/>\\\n/g' nlTest1.txt > nlTest2.txt
```

Allo stesso modo, è possibile utilizzare Perl per eseguire un'operazione simile:

```
cat nlTest1.txt | perl -p -e 's/>\s*\n/>\\\n/g' > nlTest2.txt
```

Per facilitare il caricamento dei dati dal file `nlTest2.txt` in Amazon Redshift, è stata creata una tabella a due colonne in Amazon Redshift. La prima colonna c1, è una colonna di caratteri che ha il contenuto in formato XML dal file `nlTest2.txt`. La seconda colonna c2 contiene i valori interi caricati dallo stesso file.

Dopo aver eseguito il comando `sed`, è possibile caricare correttamente i dati dal file `nlTest2.txt` in una tabella Amazon Redshift utilizzando il parametro ESCAPE.

> **Note**
>
> Quando includi il parametro ESCAPE al comando COPY, crea una sequenza di escape con un certo numero di caratteri speciali che includono il carattere di barra rovesciata (incluso newline).

```
copy t2 from 's3://mybucket/data/nlTest2.txt'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
escape
delimiter as '|';

select * from t2 order by 2;

c1           |  c2
-------------+------
<xml start>
<newline characters provide>
<line breaks at the end of each>
<line in content>
</xml>
| 1000
<xml>
</xml>       | 2000
(2 rows)
```

In modo analogo è possibile preparare i file di dati esportati da database esterni. Ad esempio, con un database Oracle, è possibile utilizzare la funzione REPLACE su ogni colonna interessata in una tabella che si desideri copiare in Amazon Redshift.

```
SELECT c1, REPLACE(c2, \n',\\n' ) as c2 from my_table_with_xml
```

Inoltre, molti strumenti di esportazione, estrazione, trasformazione e caricamento di database (ETL) che elaborano regolarmente grandi quantità di dati forniscono opzioni per specificare i caratteri di escape e i delimitatori.

Caricamento di uno shapefile in Amazon Redshift

Negli esempi seguenti viene illustrato come caricare uno shapefile Esri mediante COPY. Per ulteriori informazioni sul caricamento di shapefile, consultare Caricamento di uno shapefile in Amazon Redshift.

Caricamento di uno shapefile

I passaggi seguenti mostrano come importare OpenStreetMap dati da Amazon S3 utilizzando il comando COPY. Questo esempio presuppone che l'archivio degli shapefile norvegesi dal sito di download di Geofabrik sia stato caricato in un bucket Amazon S3 privato nella tua regione. AWS I file `.shp`, `.shx` e `.dbf` devono condividere lo stesso prefisso Amazon S3 e lo stesso nome file.

Importazione dei dati senza semplificazione

I seguenti comandi creano tabelle e importano dati che possono adattarsi alle dimensioni massime della geometria senza alcuna semplificazione. Aprire il `gis_osm_natural_free_1.shp` nel software GIS preferito e ispezionare le colonne di questo livello. Per impostazione predefinita, le colonne IDENTITY o GEOMETRY sono le prime. Quando una colonna GEOMETRY è prima, è possibile creare la tabella come illustrato di seguito.

```
CREATE TABLE norway_natural (
   wkb_geometry GEOMETRY,
   osm_id BIGINT,
   code INT,
   fclass VARCHAR,
   name VARCHAR);
```

Quando invece è la colonna IDENTITY a essere prima, è possibile creare la tabella come illustrato di seguito.

```
CREATE TABLE norway_natural_with_id (
   fid INT IDENTITY(1,1),
   wkb_geometry GEOMETRY,
   osm_id BIGINT,
   code INT,
   fclass VARCHAR,
   name VARCHAR);
```

Ora è possibile importare i dati usando COPY.

```
COPY norway_natural FROM 's3://bucket_name/shapefiles/norway/
gis_osm_natural_free_1.shp'
FORMAT SHAPEFILE
CREDENTIALS 'aws_iam_role=arn:aws:iam::123456789012:role/MyRoleName';
INFO: Load into table 'norway_natural' completed, 83891 record(s) loaded successfully
```

In alternativa, è possibile importare i dati come illustrato di seguito.

```
COPY norway_natural_with_id FROM 's3://bucket_name/shapefiles/norway/
gis_osm_natural_free_1.shp'
FORMAT SHAPEFILE
CREDENTIALS 'aws_iam_role=arn:aws:iam::123456789012:role/MyRoleName';
INFO: Load into table 'norway_natural_with_id' completed, 83891 record(s) loaded
 successfully.
```

Importazione dei dati con semplificazione

I seguenti comandi creano tabelle e provano a importare i dati che possono adattarsi alle dimensioni massime della geometria senza alcuna semplificazione. Ispezionare lo shapefile `gis_osm_water_a_free_1.shp` e creare la tabella appropriata come illustrato di seguito.

```
CREATE TABLE norway_water (
   wkb_geometry GEOMETRY,
   osm_id BIGINT,
   code INT,
   fclass VARCHAR,
   name VARCHAR);
```

Quando viene eseguito il comando COPY, viene generato un errore.

```
COPY norway_water FROM 's3://bucket_name/shapefiles/norway/gis_osm_water_a_free_1.shp'
```

```
FORMAT SHAPEFILE
CREDENTIALS 'aws_iam_role=arn:aws:iam::123456789012:role/MyRoleName';
ERROR:  Load into table 'norway_water' failed.  Check 'stl_load_errors' system table
 for details.
```

L'esecuzione della query su STL_LOAD_ERRORS ,ostra che la geometria è troppo grande.

```
SELECT line_number, btrim(colname), btrim(err_reason) FROM stl_load_errors WHERE query
 = pg_last_copy_id();
 line_number |    btrim     |                                  btrim
-------------+--------------
+-------------------------------------------------------------------------
     1184705 | wkb_geometry | Geometry size: 1513736 is larger than maximum supported
 size: 1048447
```

Per superare questo problema, viene aggiunto il parametro SIMPLIFY AUTO al comando COPY per semplificare le geometrie.

```
COPY norway_water FROM 's3://bucket_name/shapefiles/norway/gis_osm_water_a_free_1.shp'
FORMAT SHAPEFILE
SIMPLIFY AUTO
CREDENTIALS 'aws_iam_role=arn:aws:iam::123456789012:role/MyRoleName';

INFO:  Load into table 'norway_water' completed, 1989196 record(s) loaded successfully.
```

Per visualizzare le righe e le geometrie semplificate, eseguire una query su SVL_SPATIAL_SIMPLIFY.

```
SELECT * FROM svl_spatial_simplify WHERE query = pg_last_copy_id();
 query | line_number | maximum_tolerance | initial_size | simplified | final_size |
 final_tolerance
-------+-------------+-------------------+--------------+------------+------------
+----------------------
    20 |     1184704 |                -1 |      1513736 | t          |    1008808 |
 1.276386653895e-05
    20 |     1664115 |                -1 |      1233456 | t          |    1023584 |
 6.11707814796635e-06
```

L'uso di di SIMPLIFY AUTO tolleranza max con una tolleranza inferiore a quella calcolata automaticamente probabilmente restituisce un errore di importazione. In questo caso, utilizzare MAXERROR per ignorare gli errori.

```
COPY norway_water FROM 's3://bucket_name/shapefiles/norway/gis_osm_water_a_free_1.shp'
FORMAT SHAPEFILE
SIMPLIFY AUTO 1.1E-05
MAXERROR 2
CREDENTIALS 'aws_iam_role=arn:aws:iam::123456789012:role/MyRoleName';

INFO:  Load into table 'norway_water' completed, 1989195 record(s) loaded successfully.
INFO:  Load into table 'norway_water' completed, 1 record(s) could not be loaded.
 Check 'stl_load_errors' system table for details.
```

Eseguire di nuovo la query su SVL_SPATIAL_SIMPLIFY per identificare il record che COPY non è riuscito a caricare.

```
SELECT * FROM svl_spatial_simplify WHERE query = pg_last_copy_id();
 query | line_number | maximum_tolerance | initial_size | simplified | final_size |
 final_tolerance
-------+-------------+-------------------+--------------+------------+------------
+-----------------
    29 |     1184704 |           1.1e-05 |      1513736 | f          |          0 |
            0
    29 |     1664115 |           1.1e-05 |      1233456 | t          |     794432 |
     1.1e-05
```

In questo esempio, il primo record non è riuscito a adattarsi, quindi la colonna `simplified` riporta false. Il secondo record è stato caricato entro la tolleranza specificata. Tuttavia, la dimensione finale è maggiore rispetto all'utilizzo della tolleranza calcolata automaticamente senza specificare la tolleranza massima.

Caricamento da uno shapefile compresso

Il comando COPY di Amazon Redshift supporta l'importazione di dati da uno shapefile compresso. Tutti i componenti shapefile devono avere lo stesso prefisso Amazon S3 e lo stesso suffisso di compressione. Ad esempio, si supponga di voler caricare i dati dell'esempio precedente. In questo caso, i file `gis_osm_water_a_free_1.shp.gz`, `gis_osm_water_a_free_1.dbf.gz` e `gis_osm_water_a_free_1.shx.gz` devono condividere la stessa directory Amazon S3. Il comando COPY richiede l'opzione GZIP e la clausola FROM deve specificare il file compresso corretto, come illustrato di seguito.

```
COPY norway_natural FROM 's3://bucket_name/shapefiles/norway/compressed/
gis_osm_natural_free_1.shp.gz'
FORMAT SHAPEFILE
GZIP
CREDENTIALS 'aws_iam_role=arn:aws:iam::123456789012:role/MyRoleName';
INFO:  Load into table 'norway_natural' completed, 83891 record(s) loaded successfully.
```

Caricamento di dati in una tabella con un ordine di colonna diverso

Se hai una tabella che non dispone di GEOMETRY come prima colonna, è possibile mappare le colonne alla tabella di destinazione. Ad esempio, creare una tabella con osm_id specificato come prima colonna.

```
CREATE TABLE norway_natural_order (
   osm_id BIGINT,
   wkb_geometry GEOMETRY,
   code INT,
   fclass VARCHAR,
   name VARCHAR);
```

Quindi importare uno shapefile utilizzando la mappatura delle colonne.

```
COPY norway_natural_order(wkb_geometry, osm_id, code, fclass, name)
FROM 's3://bucket_name/shapefiles/norway/gis_osm_natural_free_1.shp'
FORMAT SHAPEFILE
CREDENTIALS 'aws_iam_role=arn:aws:iam::123456789012:role/MyRoleName';
INFO:  Load into table 'norway_natural_order' completed, 83891 record(s) loaded
 successfully.
```

Caricamento di dati in una tabella con una colonna geografica

Se hai una tabella con una colonna GEOGRAPHY, prima importa in una colonna GEOMETRY e quindi converti gli oggetti in oggetti GEOGRAPHY. Ad esempio, dopo aver copiato lo shapefile in una colonna GEOMETRY, modifica la tabella per aggiungere una colonna del tipo di dati GEOGRAPHY.

```
ALTER TABLE norway_natural ADD COLUMN wkb_geography GEOGRAPHY;
```

Quindi converti le geometrie in aree geografiche.

```
UPDATE norway_natural SET wkb_geography = wkb_geometry::geography;
```

Facoltativamente, è possibile eliminare la colonna GEOMETRY.

```
ALTER TABLE norway_natural DROP COLUMN wkb_geometry;
```

Comando COPY con l'opzione NOLOAD

Per convalidare i file di dati prima di caricare effettivamente i dati, utilizza l'opzione NOLOAD con il comando COPY. Amazon Redshift analizza il file di input e visualizza gli eventuali errori che si verificano. L'esempio seguente utilizza l'opzione NOLOAD e nessuna riga viene effettivamente caricata nella tabella.

```
COPY public.zipcode1
FROM 's3://mybucket/mydata/zipcode.csv'
DELIMITER ';'
IGNOREHEADER 1 REGION 'us-east-1'
NOLOAD
CREDENTIALS 'aws_iam_role=arn:aws:iam::123456789012:role/myRedshiftRole';

Warnings:
Load into table 'zipcode1' completed, 0 record(s) loaded successfully.
```

CREATE DATABASE

Crea un nuovo database.

Per creare un database, devi essere un utente con privilegi avanzati o disporre del privilegio CREATEDB. Per creare un database associato a un'integrazione zero-ETL, devi essere un superutente o disporre dei privilegi CREATEDB e CREATEUSER.

Non è possibile eseguire CREATE DATABASE all'interno di un blocco di transazione (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Sintassi

```
CREATE DATABASE database_name
```

```
[ { [ WITH ]
    [ OWNER [=] db_owner ]
    [ CONNECTION LIMIT { limit | UNLIMITED } ]
    [ COLLATE { CASE_SENSITIVE | CASE_INSENSITIVE } ]
    [ ISOLATION LEVEL { SERIALIZABLE | SNAPSHOT } ]
  }
  | { [ WITH PERMISSIONS ] FROM DATASHARE datashare_name ] OF [ ACCOUNT account_id ]
 NAMESPACE namespace_guid }
  | { FROM { { ARN '<arn>' } { WITH DATA CATALOG SCHEMA '<schema>' | WITH NO DATA
 CATALOG SCHEMA } }
             | { INTEGRATION '<integration_id>'} }
  | { IAM_ROLE  {default | 'SESSION' | 'arn:aws:iam::<account-id>:role/<role-name>' } }
```

Parametri

database_name

Nome del nuovo database. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

WITH

Parola chiave facoltativa.

OWNER

Specifica il proprietario di un database.

=

Carattere facoltativo.

db_owner

Nome utente per il proprietario del database.

CONNECTION LIMIT { limit | UNLIMITED }

Numero massimo di connessioni di database che gli utenti possono aprire contemporaneamente. Il limite non viene applicato per gli utente con privilegi avanzati. Utilizza la parola chiave UNLIMITED per consentire il numero massimo di connessioni simultanee. È possibile che venga applicato anche un limite al numero di connessioni per ciascun utente. Per ulteriori informazioni, consultare CREA UTENTE. Il valore predefinito è UNLIMITED. Per visualizzare le connessioni correnti, eseguire una query sulla vista di sistema STV_SESSIONS.

> **Note**
>
> Se si applicano entrambi i limiti di connessione utente e database, deve essere disponibile uno slot di connessione inutilizzato che rientra in entrambi i limiti quando un utente tenta di connettersi.

COLLATE { CASE_SENSITIVE | CASE_INSENSITIVE }

Una clausola che specifica se la stringa di ricerca o il confronto è CASE_SENSITIVE o CASE_INSENSITIVE. L'opzione di default è CASE_SENSITIVE.

ISOLATION LEVEL { SERIALIZABLE | SNAPSHOT }

Una clausola che specifica il livello di isolamento utilizzato quando vengono eseguite query su un database.

- Isolamento SERIALIZABLE: fornisce la serializzabilità completa per le transazioni simultanee. Per ulteriori informazioni, consulta Isolamento serializzabile.
- Isolamento SNAPSHOT: fornisce un livello di isolamento con protezione dai conflitti di aggiornamento ed eliminazione. Questa è l'impostazione predefinita per un database creato in un cluster predisposto o in uno spazio dei nomi senza server.

È possibile visualizzare con quale modello di simultaneità è in esecuzione il database come segue:

- Esegui una query sulla vista del catalogo STV_DB_ISOLATION_LEVEL. Per ulteriori informazioni, consulta STV_DB_ISOLATION_LEVEL.

```
SELECT * FROM stv_db_isolation_level;
```

- Eseguire una query sulla vista PG_DATABASE_INFO.

```
SELECT datname, datconfig FROM pg_database_info;
```

Il livello di isolamento per database appare accanto alla chiave `concurrency_model`. Un valore di 1 indica SNAPSHOT. Un valore di 2 indica SERIALIZABLE.

Nei database Amazon Redshift, sia l'isolamento SERIALIZABLE che quello SNAPSHOT sono tipi di livelli di isolamento serializzabili. Vale a dire, le letture modificabili, le letture non ripetibili e le letture fantasma vengono impedite in base allo standard SQL. Entrambi i livelli di isolamento

garantiscono che una transazione funzioni su uno snapshot di dati esistente all'inizio della transazione e che nessun'altra transazione possa modificare tale snapshot. Tuttavia, l'isolamento SNAPSHOT non fornisce la serializzabilità completa, poiché non impedisce inserimenti e aggiornamenti anomali in scrittura su diverse righe di tabella.

Lo scenario seguente illustra gli aggiornamenti anomali in scrittura con il livello di isolamento SNAPSHOT. Una tabella denominata `Numbers` contiene una colonna denominata `digits` che contiene valori `0` e `1`. L'istruzione UPDATE di ogni utente non si sovrappone all'altro utente. Tuttavia, i valori `0` e `1` vengono scambiati. L'istruzione SQL che eseguono segue questa sequenza temporale con i seguenti risultati:

Orario	Operazione utente 1	Operazione utente 2
1	BEGIN;	
2		BEGIN;

3	SELECT * FROM Numbers;
	`digits`
	`-----`
	`0`
	`1`	

4		SELECT * FROM Numbers;
		`digits`
		`------`
		`0`
		`1`

Orario	Operazione utente 1	Operazione utente 2
5	UPDATE Numbers SET digits=0 WHERE digits=1;	

6	SELECT * FROM Numbers
	`digits`
	`-`
	`-----`
	`0`
	`0`	

7	COMMIT	
8		Update Numbers SET digits=1 WHERE digits=0;

9		SELECT * FROM Numbers;
		`digits`
		`------`
		`1`
		`1`

10		COMMIT;

Orario	Operazione utente 1	Operazione utente 2

11	SELECT * FROM Numbers
	`digits`
	`-----`
	`1`
	`0`	

12		SELECT * FROM Numbers;
		`digits`
		`------`
		`1`
		`0`

Se lo stesso scenario viene eseguito utilizzando l'isolamento serializzabile, Amazon Redshift termina l'utente 2 a causa di una violazione della serializzazione e restituisce un errore `1023`. Per ulteriori informazioni, consulta Come correggere errori di isolamento serializzabile. In questo caso, solo l'utente 1 può eseguire correttamente il commit. Non tutti i carichi di lavoro richiedono un isolamento serializzabile come requisito, nel qual caso l'isolamento degli snapshot è sufficiente come livello di isolamento di destinazione per il database.

<ARN>DA ARN "

L'ARN del AWS Glue database da utilizzare per creare il database.

{SCHEMA DEL CATALOGO DATI '<schema>' | SENZA SCHEMA DEL CATALOGO DATI}

> **Note**
>
> Questo parametro è applicabile solo se il comando CREATE DATABASE utilizza anche il parametro FROM ARN.

Specifica se creare il database utilizzando uno schema per agevolare l'accesso agli oggetti da AWS Glue Data Catalog.

DALL'INTEGRAZIONE '<integration_id>'

Speciifica se creare il database utilizzando un identificatore di integrazione zero-ETL. È possibile recuperare la vista del sistema da SVV_INTEGRATION. `integration_id` Per vedere un esempio, consulta Crea database per ricevere i risultati delle integrazioni zero-ETL. Per ulteriori informazioni sulla creazione di database con integrazioni zero-ETL, consulta Creazione di database di destinazione in Amazon Redshift nella Amazon Redshift Management Guide.

IAM_ROLE { default | 'SESSION' | 'arn:aws:iam::*<Account AWS-id>*:role/*<role-name>*' }

> **Note**
>
> Questo parametro è applicabile solo se il comando CREATE DATABASE utilizza anche il parametro FROM ARN.

Se specifichi un ruolo IAM associato al cluster quando esegui il comando CREATE DATABASE, Amazon Redshift utilizzerà le credenziali del ruolo quando esegui le query sul database.

Specificare la parola chiave `default` significa utilizzare il ruolo IAM impostato come predefinito e associato al cluster.

Utilizzare `'SESSION'` se ci si connette al cluster Amazon Redshift utilizzando un'identità federata e si accede alle tabelle dallo schema esterno creato con questo comando. Per ulteriori informazioni, consulta l'argomento relativo a Utilizzo di un'identità federata per gestire l'accesso di Amazon Redshift alle risorse locali e alle tabelle esterne di Amazon Redshift Spectrum, che illustra come configurare l'identità federata.

L'Amazon Resource Name (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione. Come minimo, il ruolo IAM deve disporre dell'autorizzazione per eseguire

un'operazione LIST sul bucket Amazon S3 a cui accedere e un'operazione GET sugli oggetti Amazon S3 contenuti nel bucket. Per ulteriori informazioni sull'utilizzo di IAM_ROLE durante la creazione di un database utilizzando AWS Glue Data Catalog per le condivisioni di dati, consulta Lavorare con le condivisioni di dati gestite da Lake Formation come consumatore.

Quanto segue mostra la sintassi per la stringa di parametro IAM_ROLE per un singolo ARN.

```
IAM_ROLE 'arn:aws:iam::<aws-account-id>:role/<role-name>'
```

È possibile concatenare i ruoli in modo che il cluster possa presumere un altro ruolo IAM, possibilmente appartenente a un altro account. Puoi concatenare fino a 10 ruoli. Per ulteriori informazioni, consulta Concatenazione di ruoli IAM per Amazon Redshift Spectrum.

Per collegare a questo ruolo IAM una policy di autorizzazioni IAM simile alla seguente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessSecret",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-west-2:123456789012:secret:my-
rds-secret-VNenFy"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword",
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*"
        }
    ]
}
```

Per la procedura per creare un ruolo IAM da utilizzare con la query federata, consultare Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate.

> **Note**
>
> Non includere spazi nell'elenco dei ruoli concatenati.

Quanto segue mostra la sintassi per concatenare tre ruoli.

```
IAM_ROLE 'arn:aws:iam::<aws-account-id>:role/<role-1-name>,arn:aws:iam::<aws-account-id>:role/<role-2-name>,arn:aws:iam::<aws-account-id>:role/<role-3-name>'
```

Sintassi per l'uso di CREATE DATABASE con una unità di condivisione dati

La sintassi seguente descrive il comando CREATE DATABASE utilizzato per creare database da un datashare per condividere dati all'interno dello stesso account. AWS

```
CREATE DATABASE database_name
[ [ WITH PERMISSIONS ] FROM DATASHARE datashare_name ] OF [ ACCOUNT account_id ]
 NAMESPACE namespace_guid
```

La sintassi seguente descrive il comando CREATE DATABASE utilizzato per creare database da un datashare per la condivisione di dati tra account. AWS

```
CREATE DATABASE database_name
[ [ WITH PERMISSIONS ] FROM DATASHARE datashare_name ] OF ACCOUNT account_id
 NAMESPACE namespace_guid
```

Parametri per l'uso di CREATE DATABASE con una unità di condivisione dati

FROM DATASHARE

Una parola chiave che indica dove si trova l'unità di condivisione dati.

datashare_name

Il nome dell'unità di condivisione dati in cui viene creato il database consumer.

WITH PERMISSIONS

Specifica che il database creato dall'unità di condivisione dati richiede autorizzazioni a livello di oggetto per accedere ai singoli oggetti del database. Senza questa clausola, gli utenti o i ruoli a cui è stata concessa l'autorizzazione USAGE sul database hanno automaticamente accesso a tutti gli oggetti del database.

NAMESPACE namespace_guid

Un valore che specifica lo spazio dei nomi del produttore a cui appartiene l'unità di condivisione dati.

ACCOUNT account_id

Un valore che specifica lo spazio dei nomi del produttore a cui appartiene l'unità di condivisione dati.

Note di utilizzo per CREATE DATABASE per la condivisione di dati

In qualità di superutente del database, quando utilizzi CREATE DATABASE per creare database da datashare all'interno dell' AWS account, specifica l'opzione NAMESPACE. L'opzione ACCOUNT è facoltativa. Quando utilizzi CREATE DATABASE per creare database da condivisioni di dati tra AWS account, specifica sia ACCOUNT che NAMESPACE del produttore.

È possibile creare un solo database consumer per un'unità unità di condivisione dati in un cluster consumer. Non è possibile creare più database consumer che si riferiscono alla stessa unità di condivisione dati.

CREA DATABASE da AWS Glue Data Catalog

Per creare un database utilizzando un ARN di AWS Glue database, specifica l'ARN nel comando CREATE DATABASE.

```
CREATE DATABASE sampledb FROM ARN <glue-database-arn> WITH NO DATA CATALOG SCHEMA;
```

Facoltativamente, puoi anche fornire un valore nel parametro IAM_ROLE. Per ulteriori informazioni sui parametri e sui valori accettati, consulta Parametri.

Di seguito sono riportati esempi che dimostrano come creare un database da un ARN utilizzando un ruolo IAM.

```
CREATE DATABASE sampledb FROM ARN <glue-database-arn> WITH NO DATA CATALOG SCHEMA
 IAM_ROLE <iam-role-arn>
```

```
CREATE DATABASE sampledb FROM ARN <glue-database-arn> WITH NO DATA CATALOG SCHEMA
 IAM_ROLE default;
```

È inoltre possibile creare un database utilizzando uno SCHEMA DEL CATALOGO DATI.

```
CREATE DATABASE sampledb FROM ARN <glue-database-arn> WITH DATA CATALOG SCHEMA
 <sample_schema> IAM_ROLE default;
```

Crea database per ricevere i risultati delle integrazioni zero-ETL

Per creare un database utilizzando un'identità di integrazione zero-ETL, specifica nel comando CREATE DATABASE. `integration_id`

```
CREATE DATABASE destination_db_name FROM INTEGRATION 'integration_id';
```

Ad esempio, per prima cosa, recuperate gli ID di integrazione da SVV_INTEGRATION;

```
SELECT integration_id FROM SVV_INTEGRATION;
```

Quindi utilizzate uno degli id di integrazione recuperati per creare il database che riceve integrazioni zero-ETL.

```
CREATE DATABASE sampledb FROM INTEGRATION 'a1b2c3d4-5678-90ab-cdef-EXAMPLE11111';
```

Limiti di CREATE DATABASE

Amazon Redshift applica i seguenti limiti per i database:

- Massimo 60 database definiti dall'utente per cluster.
- Massimo 127 byte per un nome del database.
- Il nome di un database non può essere una parola riservata.

Confronto di database

Il confronto è un insieme di regole che definisce il modo in cui il motore di database confronta e ordina i dati di tipo carattere in SQL. Il confronto senza distinzione tra maiuscole e minuscole è il tipo di confronto più utilizzato. Amazon Redshift utilizza il confronto senza distinzione tra maiuscole e minuscole per facilitare la migrazione da altri sistemi di data warehouse. Con il supporto nativo del confronto senza distinzione tra maiuscole e minuscole, Amazon Redshift continua a utilizzare importanti metodi di regolazione o ottimizzazione, come le chiavi di distribuzione, le chiavi di ordinamento o la scansione con intervallo limitato.

La clausola COLLATE specifica il confronto di default per tutte le colonne CHAR e VARCHAR nel database. Se è specificato CASE_INSENSITIVE, tutte le colonne CHAR o VARCHAR utilizzano confronto senza distinzione tra maiuscole e minuscole Per informazioni sul confronto, consultare Sequenze di regole di confronto.

I dati inseriti o importati nelle colonne senza distinzione tra maiuscole e minuscole manterranno il loro formato maiuscole/minuscole originale. Ma tutte le operazioni di stringa basate su confronto, inclusi l'ordinamento e il raggruppamento, non fanno distinzione tra maiuscole e minuscole. Anche le operazioni di corrispondenza dei modelli come predicati LIKE, SIMILAR TO e funzioni di espressione regolare non fanno distinzione tra maiuscole e minuscole.

Le seguenti operazioni SQL supportano la semantica di confronto applicabile:

- Operatori di confronto: =, <>, <, <=, >, >=.
- Operatore LIKE
- Clausola ORDER BY
- Clausole GROUP BY
- Funzioni di aggregazione che utilizzano il confronto di stringhe, come MIN e MAX e LISTAGG
- Funzioni finestra, ad esempio clausole PARTITION BY e clausole ORDER BY
- Funzioni scalari greatest() e least(), STRPOS(), REGEXP_COUNT(), REGEXP_REPLACE(), REGEXP_INSTR(), REGEXP_SUBSTR()
- Clausola separata
- UNION, INTERSECT ed EXCEPT
- IN LIST

Per le query esterne, incluse le query federate di Amazon Redshift Spectrum e Aurora PostgreSQL, il confronto della colonna VARCHAR o CHAR è uguale al confronto corrente a livello di database.

Nell'esempio seguente viene eseguita una query su una tabella Amazon Redshift Spectrum:

```
SELECT ci_varchar FROM spectrum.test_collation
WHERE ci_varchar = 'AMAZON';

ci_varchar
----------
amazon
Amazon
AMAZON
AmaZon
(4 rows)
```

Per informazioni su come creare le tabelle mediante il confronto di database, consultare CREATE TABLE.

Per ulteriori informazioni sulla funzione COLLATE, consultare Funzione COLLATE.

Limitazioni del confronto di database

Di seguito sono elencate le limitazioni nell'utilizzo del confronto di database in Amazon Redshift:

- Tutte le tabelle o le viste di sistema, incluse le tabelle dei cataloghi PG e le tabelle di sistema di Amazon Redshift, fanno distinzione tra maiuscole e minuscole.
- Quando il database consumer e il database producer hanno regole di confronto diverse a livello di database, Amazon Redshift non supporta query tra database e tra cluster.
- Amazon Redshift non supporta il confronto senza distinzione tra maiuscole e minuscole nella query sul solo nodo principale.

  Nell'esempio seguente viene mostrata una query senza distinzione tra maiuscole e minuscole non supportata e l'errore restituito da Amazon Redshift:

  ```
  SELECT collate(usename, 'case_insensitive') FROM pg_user;
  ERROR:  Case insensitive collation is not supported in leader node only query.
  ```

- Amazon Redshift non supporta l'interazione tra le colonne con distinzione o senza distinzione tra maiuscole e minuscole, ad esempio operazioni di confronto, funzione, join o set.

Negli esempi seguenti vengono mostrati gli errori restituiti quando interagiscono colonne con distinzione e senza distinzione tra maiuscole e minuscole:

```
CREATE TABLE test
  (ci_col varchar(10) COLLATE case_insensitive,
   cs_col varchar(10) COLLATE case_sensitive,
   cint int,
   cbigint bigint);
```

```
SELECT ci_col = cs_col FROM test;
ERROR:  Query with different collations is not supported yet.
```

```
SELECT concat(ci_col, cs_col) FROM test;
ERROR:  Query with different collations is not supported yet.
```

```
SELECT ci_col FROM test UNION SELECT cs_col FROM test;
ERROR:  Query with different collations is not supported yet.
```

```
SELECT * FROM test a, test b WHERE a.ci_col = b.cs_col;
ERROR:  Query with different collations is not supported yet.
```

```
Select Coalesce(ci_col, cs_col) from test;
ERROR:  Query with different collations is not supported yet.
```

```
Select case when cint > 0 then ci_col else cs_col end from test;
ERROR:  Query with different collations is not supported yet.
```

- Amazon Redshift non supporta il confronto per il tipo di dati SUPER. La creazione di colonne SUPER nei database senza distinzione tra maiuscole e minuscole e le interazioni tra colonne SUPER e colonne senza distinzione tra maiuscole e minuscole non sono supportate.

  Nell'esempio seguente viene creata una tabella con SUPER come tipo di dati nel database senza distinzione tra maiuscole e minuscole:

```
CREATE TABLE super_table (a super);
ERROR: SUPER column is not supported in case insensitive database.
```

Nell'esempio seguente viene eseguita una query sui dati con una stringa senza distinzione tra maiuscole e minuscole confrontando i dati SUPER:

```
CREATE TABLE test_super_collation
(s super, c varchar(10) COLLATE case_insensitive, i int);
```

```
SELECT s = c FROM test_super_collation;
ERROR:  Coercing from case insensitive string to SUPER is not supported.
```

Per far funzionare queste query, utilizzare la funzione COLLATE per convertire il confronto di una colonna in modo che corrisponda all'altra. Per ulteriori informazioni, consulta Funzione COLLATE.

Esempi

Creazione di un database

Nell'esempio seguente viene creato un database denominato TICKIT e ne viene assegnata la proprietà all'utente DWUSER:

```
create database tickit
with owner dwuser;
```

Per visualizzare i dettagli sui database, eseguire una query sulla tabella di catalogo PG_DATABASE_INFO.

```
select datname, datdba, datconnlimit
from pg_database_info
where datdba > 1;

 datname     | datdba | datconnlimit
-------------+--------+-------------
 admin       |    100 | UNLIMITED
 reports     |    100 | 100
 tickit      |    100 | 100
```

L'esempio seguente crea un database denominato **`sampledb`** con livello di isolamento SNAPSHOT.

```
CREATE DATABASE sampledb ISOLATION LEVEL SNAPSHOT;
```

Nell'esempio seguente viene creato il database sales_db dall'unità di condivisione dati salesshare.

```
CREATE DATABASE sales_db FROM DATASHARE salesshare OF NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

Esempi di confronto di database

Creazione di un database senza distinzione tra maiuscole e minuscole

Nell'esempio seguente viene creato il database `sampledb`, la tabella `T1` e sono inseriti i dati nella tabella `T1`.

```
create database sampledb collate case_insensitive;
```

Esegui la connessione al nuovo database che hai appena creato usando il client SQL. Se usi l'Editor di query Amazon Redshift v2, scegli `sampledb` in Editor. Se usi RSQL, esegui un comando come il seguente.

```
\connect sampledb;
```

```
CREATE TABLE T1 (
  col1 Varchar(20) distkey sortkey
);
```

```
INSERT INTO T1 VALUES ('bob'), ('john'), ('Mary'), ('JOHN'), ('Bob');
```

Quindi la query trova i risultati con `John`.

```
SELECT * FROM T1 WHERE col1 = 'John';

 col1
 ------
 john
 JOHN
(2 row)
```

Ordinamento senza distinzione tra maiuscole e minuscole

Nell'esempio seguente è mostrato l'ordinamento senza distinzione tra maiuscole e minuscole con la tabella T1. L'ordinamento di Bob e bob o John e john non è deterministico perché sono uguali nella colonna senza distinzione tra maiuscole e minuscole.

```
SELECT * FROM T1 ORDER BY 1;

 col1
 ------
 bob
 Bob
 JOHN
 john
 Mary
(5 rows)
```

Analogamente, nell'esempio seguente è mostrato l'ordinamento senza distinzione tra maiuscole e minuscole con la clausola GROUP BY. Bob e bob sono uguali e appartengono allo stesso gruppo. È non deterministico ciò che si presenta nel risultato.

```
SELECT col1, count(*) FROM T1 GROUP BY 1;

 col1 | count
 -----+------
 Mary |  1
 bob  |  2
 JOHN |  2
(3 rows)
```

Esecuzione di query con una funzione finestra su colonne senza distinzione tra maiuscole e minuscole

Nell'esempio seguente viene eseguita una query su una funzione finestra senza distinzione tra maiuscole e minuscole.

```
SELECT col1, rank() over (ORDER BY col1) FROM T1;

 col1 | rank
 -----+------
 bob  |   1
 Bob  |   1
 john |   3
 JOHN |   3
```

```
 Mary |   5
(5 rows)
```

Esecuzione di query con la parola chiave DISTINCT

Nell'esempio seguente viene eseguita una query sulla tabella T1 con la parola chiave DISTINCT.

```
SELECT DISTINCT col1 FROM T1;

 col1
 ------
 bob
 Mary
 john
(3 rows)
```

Esecuzione di query con la clausola UNION

Nell'esempio seguente sono mostrati i risultati dell'UNION delle tabelle T1 e T2.

```
CREATE TABLE T2 AS SELECT * FROM T1;
```

```
SELECT col1 FROM T1 UNION SELECT col1 FROM T2;

 col1
 ------
 john
 bob
 Mary
(3 rows)
```

CREARE DATASHARE

Crea una nuova unità di condivisione dati nel database corrente. Il proprietario di questa unità di condivisione dati è l'emittente del comando CREATE DATASHARE.

Amazon Redshift associa ogni unità di condivisione dati a un unico database Amazon Redshift. È possibile aggiungere oggetti all'unità di condivisione dati solo da quel database associato. È possibile creare più unità di condivisione dati sullo stesso database Amazon Redshift.

Per informazioni sulle unità di condivisione dati, consultare Gestione delle attività di condivisione dati.

Per visualizzare informazioni sulle unità di condivisione dati, utilizzare SHOW DATASHARES.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per CREATE DATASHARE:

- Superuser
- Utenti con il privilegio CREATE DATASHARE
- Proprietario del database

Sintassi

```
CREATE DATASHARE datashare_name
[[SET] PUBLICACCESSIBLE [=] TRUE | FALSE ];
```

Parametri

datashare_name

Il nome dell'unità di condivisione dati. Il nome dell'unità di condivisione dati deve essere univoco nello spazio dei nomi del cluster.

[[SET] PUBLICACCESSIBLE]

Una clausola che specifica se una unità di condivisione dati può essere condivisa nei cluster accessibili pubblicamente.

Il valore predefinito per `SET PUBLICACCESSIBLE` è `FALSE`.

Note per l'utilizzo

Per impostazione predefinita, il proprietario dell'unità di condivisione dati possiede solo la condivisione e non gli oggetti all'interno della condivisione.

Solo gli utenti con privilegi avanzati e il proprietario del database possono utilizzare CREATE DATASHARE e delegare i privilegi ALTER ad altri utenti o gruppi.

Esempi

Nell'esempio seguente viene creata l'unità di condivisione dati `salesshare`.

```
CREATE DATASHARE salesshare;
```

Nell'esempio seguente viene creata l'unità di condivisione dati `demoshare` gestita da AWS Data Exchange .

```
CREATE DATASHARE demoshare SET PUBLICACCESSIBLE TRUE, MANAGEDBY ADX;
```

CREATE EXTERNAL FUNCTION

Crea una funzione scalare definita dall'utente (UDF) basata su Amazon AWS Lambda Redshift. Per ulteriori informazioni sulle funzioni Lambda definite dall'utente, consultare Creazione di una funzione Lambda definita dall'utente scalare.

Privilegi richiesti

Di seguito sono riportati i privilegi necessari per CREATE EXTERNAL FUNCTION:

- Superuser
- Utenti con il privilegio CREATE [OR REPLACE] EXTERNAL FUNCTION

Sintassi

```
CREATE [ OR REPLACE ] EXTERNAL FUNCTION external_fn_name ( [data_type] [, ...] )
RETURNS data_type
{ VOLATILE | STABLE }
LAMBDA 'lambda_fn_name'
IAM_ROLE { default | ‘arn:aws:iam::<Account AWS-id>:role/<role-name>’
RETRY_TIMEOUT milliseconds
MAX_BATCH_ROWS count
MAX_BATCH_SIZE size [ KB | MB ];
```

Parametri

OR REPLACE

Una clausola che specifica che se una funzione con stesso nome e tipi di dati degli argomenti di input o firma è già esistente, la funzione esistente viene sostituita. Puoi sostituire una funzione solo con una nuova funzione che definisce un set identico di tipi di dati. Devi essere un utente con privilegi avanzati per sostituire una funzione.

Se definisci una funzione con lo stesso nome di una funzione esistente ma con una firma diversa, viene creata una nuova funzione. In altre parole, il nome della funzione è sottoposto a overload. Per ulteriori informazioni, consulta Overload dei nomi delle funzioni.

external_fn_name

Il nome della funzione esterna. Se si specifica un nome schema (come myschema.myfunction), la funzione viene creata utilizzando lo schema specificato. Altrimenti, la funzione viene creata nello schema corrente. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

Consigliamo di assegnare un prefisso `f_` ai nomi di tutte le funzioni definite dall'utente. Amazon Redshift riserva il prefisso `f_` per i nomi delle funzioni definite dall'utente. Utilizzando il prefisso `f_`, è possibile assicurarsi che il nome della funzione non sia in conflitto con i nomi delle funzioni SQL predefinite di Amazon Redshift correnti e future. Per ulteriori informazioni, consulta Denominazione delle funzioni definite dall'utente.

data_type

Il tipo di dati per gli argomenti di input. Per ulteriori informazioni, consulta Tipi di dati.

RETURNS data_type

Il tipo di dati del valore restituito dalla funzione. Il tipo di dati RETURNS può essere qualsiasi tipo di dati standard Amazon Redshift. Per ulteriori informazioni, consulta Tipi di dati delle funzioni definite dall'utente Python.

VOLATILE | STABILE

Informa l'ottimizzatore di query circa la volatilità della funzione.

Per ottenere la migliore ottimizzazione, etichettare la funzione con la categoria di volatilità più rigida valida. In ordine di rigidità, a partire dalla meno rigida, le categorie di volatilità sono le seguenti:

- VOLATILE
- STABLE

VOLATILE

Dati gli stessi argomenti, la funzione può restituire risultati diversi su chiamate successive, anche per le righe in una singola istruzione. L'ottimizzatore di query non può fare alcuna ipotesi sul comportamento di una funzione volatile. Una query che utilizza una funzione volatile deve rivalutare la funzione per ogni input.

STABLE

Dati gli stessi argomenti, la funzione è garantita per restituire gli stessi risultati nelle chiamate successive elaborate all'interno di una singola istruzione. La funzione può restituire risultati diversi se richiamati in istruzioni diverse. Questa categoria consente all'ottimizzatore di ridurre il numero di volte in cui la funzione viene chiamata all'interno di una singola istruzione.

Nota che se la severità scelta non è valida per la funzione, c'è il rischio che l'ottimizzatore salti alcune chiamate in base a questa severità. Ciò può comportare un set di risultati errato.

La clausola IMMUTABLE non è attualmente supportata per le UDF Lambda.

LAMBDA 'lambda_fn_name'

Il nome della funzione richiamata da Amazon Redshift.

Per i passaggi per creare una AWS Lambda funzione, consulta Creare una funzione Lambda con la console nella Guida per gli AWS Lambda sviluppatori.

Per informazioni sulle autorizzazioni richieste per la funzione Lambda, consultare Autorizzazioni AWS Lambda nella Guida per gli sviluppatori di AWS Lambda .

IAM_ROLE { default | 'arn:aws:iam::*<Account AWS-id>*:ruolo/*<role-name>*'

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando CREATE EXTERNAL FUNCTION.

L'Amazon Resource Name (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione. Il comando CREATE EXTERNAL FUNCTION è autorizzato per il richiamo delle funzioni Lambda tramite questo ruolo IAM. Se il cluster possiede un ruolo IAM esistente con autorizzazioni per richiamare funzioni Lambda collegate, è possibile sostituire l'ARN del ruolo. Per ulteriori informazioni, consulta Configurazione del parametro di autorizzazione per le funzioni Lambda definite dall'utente.

Di seguito è mostrata la sintassi del parametro IAM_ROLE.

```
IAM_ROLE 'arn:aws:iam::aws-account-id:role/role-name'
```

RETRY_TIMEOUT millisecondi

La quantità di tempo totale, espressa in millisecondi, utilizzata da Amazon Redshift per i ritardi nei backoff dei tentativi.

Invece di riprovare immediatamente per eventuali query non riuscite, Amazon Redshift esegue i backoff e attende un certo periodo di tempo tra i tentativi. Quindi Amazon Redshift riprova la richiesta di eseguire nuovamente la query non riuscita fino a quando la somma di tutti i ritardi è uguale o superiore al valore RETRY_TIMEOUT specificato. Il valore di default è 20.000 millisecondi.

Quando viene richiamata una funzione Lambda, Amazon Redshift prova a eseguire nuovamente le query che ricevono errori come `TooManyRequestsException`, `EC2ThrottledException` e `ServiceException`.

È possibile impostare il parametro RETRY_TIMEOUT su 0 millisecondi per evitare nuovi tentativi per una funzione Lambda definita dall'utente.

MAX_BATCH_ROWS count

Numero massimo di righe che Amazon Redshift invia in una singola richiesta batch per una singola chiamata lambda.

Il valore minimo per questo parametro è 1. Il valore massimo è INT_MAX o 2.147.483.647.

Questo parametro è facoltativo. Il valore predefinito è INT_MAX o 2.147.483.647.

MAX_BATCH_SIZE size [KB | MB]

La dimensione massima del payload di dati che Amazon Redshift invia in una singola richiesta batch per una singola chiamata lambda.

Il valore minimo per questo parametro è 1 KB. Il valore massimo è 5 MB.

Il valore predefinito di questo parametro è 5 MB.

KB e MB sono facoltativi. Se non si imposta l'unità di misura, Amazon Redshift utilizza per impostazione predefinita KB.

Note per l'utilizzo

Quando crei UDF Lambda, tieni presente quanto segue:

- L'ordine delle chiamate alla funzione Lambda sugli argomenti di input non è fisso o garantito. Può variare tra le istanze di esecuzione delle query, a seconda della configurazione del cluster.

- Non è garantito che le funzioni vengano applicate a ciascun argomento di input una sola volta. L'interazione tra Amazon Redshift e AWS Lambda potrebbe portare a chiamate ripetitive con gli stessi input.

Esempi

Di seguito sono riportati alcuni esempi di utilizzo di funzioni Lambda definite dall'utente scalari.

Esempio di funzione Lambda definita dall'utente scalare che utilizza una funzione Lambda Node.js

L'esempio seguente crea una funzione esterna denominata `exfunc_sum` che utilizza due numeri interi come argomenti di input. Questa funzione restituisce la somma come output intero. Il nome della funzione Lambda da chiamare è `lambda_sum`. La lingua utilizzata per questa funzione Lambda è Node.js 12.x. Assicurarsi di specificare il ruolo IAM. L'esempio utilizza `'arn:aws:iam::123456789012:user/johndoe'` come ruolo IAM.

```
CREATE EXTERNAL FUNCTION exfunc_sum(INT,INT)
RETURNS INT
VOLATILE
LAMBDA 'lambda_sum'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-Exfunc-Test';
```

La funzione Lambda accetta il payload della richiesta ed esegue l'iterazione su ogni riga. Tutti i valori di una singola riga vengono aggiunti per calcolare la somma per quella riga, che viene salvata nell'array di risposte. Il numero di righe nell'array dei risultati è simile al numero di righe ricevute nel payload della richiesta.

Perché sia riconosciuto dalla funzione esterna, il payload della risposta JSON deve avere i dati di risultato nel campo 'results'. Il campo argomenti nella richiesta inviata alla funzione Lambda contiene il payload dei dati. In caso di richiesta batch, è possibile avere più righe nel payload dei dati. La seguente funzione Lambda itera su tutte le righe nel payload dei dati della richiesta. Inoltre, itera individualmente su tutti i valori all'interno di una singola riga.

```
exports.handler = async (event) => {
    // The 'arguments' field in the request sent to the Lambda function contains the
 data payload.
    var t1 = event['arguments'];

    // 'len(t1)' represents the number of rows in the request payload.
```

```
    // The number of results in the response payload should be the same as the number
 of rows received.
    const resp = new Array(t1.length);

    // Iterating over all the rows in the request payload.
    for (const [i, x] of t1.entries())
    {
        var sum = 0;
        // Iterating over all the values in a single row.
        for (const y of x) {
            sum = sum + y;
        }
        resp[i] = sum;
    }
    // The 'results' field should contain the results of the lambda call.
    const response = {
        results: resp
    };
    return JSON.stringify(response);
};
```

Nell'esempio seguente viene chiamata la funzione esterna con valori letterali.

```
select exfunc_sum(1,2);
exfunc_sum
------------
 3
(1 row)
```

Nell'esempio seguente viene creata una tabella denominata t_sum con due colonne, c1 e c2, del tipo di dati intero e vengono inserite due righe di dati. Quindi la funzione esterna viene chiamata passando i nomi delle colonne di questa tabella. Le due righe della tabella vengono inviate in una richiesta batch nel payload della richiesta come un singolo richiamo Lambda.

```
CREATE TABLE t_sum(c1 int, c2 int);
INSERT INTO t_sum VALUES (4,5), (6,7);
SELECT exfunc_sum(c1,c2) FROM t_sum;
 exfunc_sum
---------------
 9
 13
(2 rows)
```

Esempio di funzione Lambda definita dall'utente scalare che utilizza l'attributo RETRY_TIMEOUT

Nella sezione seguente viene fornito un esempio di come utilizzare l'attributo RETRY_TIMEOUT nelle funzioni Lambda definite dall'utente.

AWS Lambda le funzioni hanno limiti di concorrenza che puoi impostare per ogni funzione. Per ulteriori informazioni sui limiti di concorrenza, consulta Gestire la concorrenza per una funzione Lambda nella Guida per gli AWS Lambda sviluppatori e il post Managing AWS Lambda Function Concurrency sul blog di Compute. AWS

Quando il numero di richieste servite da una funzione Lambda definita dall'utente supera i limiti di simultaneità, le nuove richieste ricevono l'errore `TooManyRequestsException`. La funzione Lambda definita dall'utente riprova su questo errore fino a quando la somma di tutti i ritardi tra le richieste inviate alla funzione Lambda è uguale o superiore al valore RETRY_TIMEOUT impostato. Il valore di default di RETRY_TIMEOUT è 20.000 millisecondi.

L'esempio seguente crea una funzione Lambda denominata `exfunc_sleep_3`. Questa funzione prende in carico il payload della richiesta, itera su ogni riga e converte l'input in maiuscolo. Quindi si interrompe per 3 secondi e restituisce il risultato. Il linguaggio utilizzato per questa funzione Lambda è Python 3.8.

Il numero di righe nell'array dei risultati è simile al numero di righe ricevute nel payload della richiesta. Perché sia riconosciuto dalla funzione esterna, il payload della risposta JSON deve avere i dati di risultato nel campo `results`. Il campo `arguments` nella richiesta inviata alla funzione Lambda contiene il payload dei dati. In caso di richiesta batch, è possibile avere più righe nel payload dei dati.

Il limite di simultaneità per questa funzione è impostato specificatamente su 1 nella simultaneità riservata per dimostrare l'utilizzo dell'attributo RETRY_TIMEOUT. Quando l'attributo è impostato su 1, la funzione Lambda può servire solo una richiesta alla volta.

```
import json
import time
def lambda_handler(event, context):
    t1 = event['arguments']
    # 'len(t1)' represents the number of rows in the request payload.
    # The number of results in the response payload should be the same as the number of
 rows received.
    resp = [None]*len(t1)

    # Iterating over all rows in the request payload.
```

```
    for i, x in enumerate(t1):
        # Iterating over all the values in a single row.
        for j, y in enumerate(x):
            resp[i] = y.upper()

    time.sleep(3)
    ret = dict()
    ret['results'] = resp
    ret_json = json.dumps(ret)
    return ret_json
```

Di seguito, due esempi aggiuntivi illustrano l'attributo RETRY_TIMEOUT. Ognuno di essi richiama una singola funzione Lambda definita dall'utente. Durante il richiamo della funzione Lambda, ogni esempio esegue la stessa query SQL per richiamare la funzione Lambda da due sessioni di database simultanee contemporaneamente. Quando la prima query che richiama la funzione Lambda definita dall'utente viene servita dalla funzione, la seconda query riceve l'errore `TooManyRequestsException`. Questo risultato si verifica perché si imposta in modo specifico la simultaneità riservata nella funzione definita dall'utente su 1. Per informazioni su come configurare la simultaneità riservata per le funzioni Lambda, consultare Configurazione della simultaneità riservata.

Nel primo esempio che segue, l'attributo RETRY_TIMEOUT per la funzione Lambda definita dall'utente viene impostato su 0 millisecondi. Se la richiesta Lambda riceve una qualsiasi eccezione dalla funzione Lambda, Amazon Redshift non esegue alcun nuovo tentativo. Questo risultato si verifica perché l'attributo RETRY_TIMEOUT è impostato su 0.

```
CREATE OR REPLACE EXTERNAL FUNCTION exfunc_upper(varchar)
RETURNS varchar
VOLATILE
LAMBDA 'exfunc_sleep_3'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-Exfunc-Test'
RETRY_TIMEOUT 0;
```

Con RETRY_TIMEOUT impostato su 0, è possibile eseguire le seguenti due query da sessioni di database separate e visualizzare risultati diversi.

La prima query SQL che utilizza la funzione Lambda definita dall'utente viene eseguita correttamente.

```
select exfunc_upper('Varchar');
 exfunc_upper
 --------------
```

```
 VARCHAR
(1 row)
```

La seconda query, eseguita contemporaneamente da una sessione di database separata, riceve l'errore `TooManyRequestsException`.

```
select exfunc_upper('Varchar');
ERROR:  Rate Exceeded.; Exception: TooManyRequestsException; ShouldRetry: 1
DETAIL:
-----------------------------------------------
error:  Rate Exceeded.; Exception: TooManyRequestsException; ShouldRetry: 1
code:      32103
context:query:     0
location:  exfunc_client.cpp:102
process:   padbmaster [pid=26384]
-----------------------------------------------
```

Nel primo esempio che segue, l'attributo RETRY_TIMEOUT per la funzione Lambda definita dall'utente viene impostato su 3.000 millisecondi. Anche se la seconda query viene eseguita contemporaneamente, la funzione Lambda definita dall'utente tenta di nuovo fino a quando il ritardo totale è di 3.000 millisecondi. Pertanto, entrambe le query vengono eseguite correttamente.

```
CREATE OR REPLACE EXTERNAL FUNCTION exfunc_upper(varchar)
RETURNS varchar
VOLATILE
LAMBDA 'exfunc_sleep_3'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-Exfunc-Test'
RETRY_TIMEOUT 3000;
```

Con RETRY_TIMEOUT impostato su 3.000 millisecondi, è possibile eseguire le seguenti due query da sessioni di database separate e visualizzare gli stessi risultati.

La prima query SQL che esegue la funzione Lambda definita dall'utente viene eseguita correttamente.

```
select exfunc_upper('Varchar');
 exfunc_upper
 --------------
 VARCHAR
(1 row)
```

La seconda query viene eseguita contemporaneamente e la funzione Lambda definita dall'utente prova di nuovo fino a quando il ritardo totale è di 3.000 millisecondi.

```
select exfunc_upper('Varchar');
 exfunc_upper
--------------
 VARCHAR
(1 row)
```

Esempio di funzione Lambda definita dall'utente scalare che utilizza una funzione Lambda Python

L'esempio seguente crea una funzione esterna denominata `exfunc_multiplication` e che moltiplica i numeri e restituisce un numero intero. Questo esempio incorpora il successo e i campi `error_msg` nella risposta Lambda. Il campo di successo è impostato su false quando c'è un overflow di numeri interi nel risultato della moltiplicazione e il messaggio `error_msg` è impostato su `Integer multiplication overflow`. La funzione `exfunc_multiplication` accetta tre numeri interi come argomenti di input e restituisce la somma come output intero.

Il nome della funzione Lambda richiamata è `lambda_multiplication`. Il linguaggio utilizzato per questa funzione Lambda è Python 3.8. Assicurarsi di specificare il ruolo IAM.

```
CREATE EXTERNAL FUNCTION exfunc_multiplication(int, int, int)
RETURNS INT
VOLATILE
LAMBDA 'lambda_multiplication'
IAM_ROLE 'arn:aws:iam::123456789012:role/Redshift-Exfunc-Test';
```

La funzione Lambda accetta il payload della richiesta ed esegue l'iterazione su ogni riga. Tutti i valori di una singola riga vengono moltiplicati per calcolare il risultato di tale riga, che viene salvato nell'elenco delle risposte. In questo esempio viene utilizzato un valore di successo booleano impostato su true per impostazione predefinita. Se il risultato della moltiplicazione per una riga ha un overflow intero, il valore di successo è impostato su false. Quindi il ciclo di iterazione si interrompe.

Durante la creazione del payload di risposta, se il valore di successo è false, la seguente funzione Lambda aggiunge il campo `error_msg` nel payload. Imposta anche il messaggio di errore su `Integer multiplication overflow`. Se il valore di successo è true, i dati dei risultati vengono aggiunti nel campo dei risultati. Il numero di righe nell'array dei risultati, se presente, è simile al numero di righe ricevute nel payload della richiesta.

Il campo argomenti nella richiesta inviata alla funzione Lambda contiene il payload dei dati. In caso di richiesta batch, è possibile avere più righe nel payload dei dati. La seguente funzione Lambda itera su tutte le righe nel payload dei dati richiesta e itera individualmente su tutti i valori all'interno di una singola riga.

```
import json
def lambda_handler(event, context):
    t1 = event['arguments']
    # 'len(t1)' represents the number of rows in the request payload.
    # The number of results in the response payload should be the same as the number of
 rows received.
    resp = [None]*len(t1)

    # By default success is set to 'True'.
    success = True
    # Iterating over all rows in the request payload.
    for i, x in enumerate(t1):
        mul = 1
        # Iterating over all the values in a single row.
        for j, y in enumerate(x):
            mul = mul*y

        # Check integer overflow.
        if (mul >= 9223372036854775807 or mul <= -9223372036854775808):
            success = False
            break
        else:
            resp[i] = mul
    ret = dict()
    ret['success'] = success
    if not success:
        ret['error_msg'] = "Integer multiplication overflow"
    else:
        ret['results'] = resp
    ret_json = json.dumps(ret)

    return ret_json
```

Nell'esempio seguente viene chiamata la funzione esterna con valori letterali.

```
SELECT exfunc_multiplication(8, 9, 2);
  exfunc_multiplication
--------------------------
```

```
          144
(1 row)
```

Nell'esempio seguente viene creata una tabella denominata t_multi con tre colonne, c1, c2 e c3, del tipo di dati intero. La funzione esterna viene chiamata passando i nomi delle colonne di questa tabella. I dati vengono inseriti in modo tale da causare l'overflow di interi per mostrare come l'errore viene propagato.

```
CREATE TABLE t_multi (c1 int, c2 int, c3 int);
INSERT INTO t_multi VALUES (2147483647, 2147483647, 4);
SELECT exfunc_multiplication(c1, c2, c3) FROM t_multi;
DETAIL:
  -----------------------------------------------
  error:  Integer multiplication overflow
  code:      32004context:
  context:
  query:     38
  location:  exfunc_data.cpp:276
  process:   query2_16_38 [pid=30494]
  -----------------------------------------------
```

CREATE EXTERNAL SCHEMA

Crea un nuovo schema esterno nel database corrente. È possibile utilizzare questo schema esterno per connettersi al database Amazon RDS for PostgreSQL o al database Amazon Aurora Edizione compatibile con PostgreSQL. Puoi anche creare uno schema esterno che faccia riferimento a un database in un catalogo di dati esterno come AWS Glue Athena o a un database in un metastore Apache Hive, come Amazon EMR.

Il proprietario di questo schema è l'emittente del comando CREATE EXTERNAL SCHEMA. Per trasferire la proprietà di uno schema esterno, utilizza ALTER SCHEMA per cambiare il proprietario. Usa il comando GRANT per concedere l'accesso allo schema ad altri utenti o gruppi di utenti.

Non puoi utilizzare i comandi GRANT o REVOKE per le autorizzazioni su una tabella esterna. Puoi invece concedere o revocare le autorizzazioni per lo schema esterno.

> **Note**
>
> Se attualmente si dispone di tabelle esterne Redshift Spectrum nel catalogo dati di Amazon Athena, è possibile migrare tale catalogo di Athena a un AWS Glue Data Catalog. Per

utilizzare AWS Glue Data Catalog con Redshift Spectrum, potrebbe essere necessario modificare le policy AWS Identity and Access Management (IAM). Per ulteriori informazioni, consulta Aggiornamento al AWS Glue Data Catalog nella Guida per l'utente di Athena.

Per visualizzare i dettagli degli schemi esterni, eseguire una query sulla vista di sistema SVV_EXTERNAL_SCHEMAS.

Sintassi

La sintassi seguente descrive il comando CREATE EXTERNAL SCHEMA utilizzato per fare riferimento ai dati utilizzando un catalogo dati esterno. Per ulteriori informazioni, consulta Esecuzione di query sui dati esterni utilizzando Amazon Redshift Spectrum.

```
CREATE EXTERNAL SCHEMA [IF NOT EXISTS] local_schema_name
FROM { [ DATA CATALOG ] | HIVE METASTORE | POSTGRES | MYSQL | KINESIS | MSK |
 REDSHIFT }
[ DATABASE 'database_name' ]
[ SCHEMA 'schema_name' ]
[ REGION 'aws-region' ]
[ URI 'hive_metastore_uri' [ PORT port_number ] ]
IAM_ROLE { default | 'SESSION' | 'arn:aws:iam::<Account AWS-id>:role/<role-name>' }
[ SECRET_ARN 'ssm-secret-arn' ]
[ AUTHENTICATION { none | iam } ]
[ CLUSTER_ARN 'arn:aws:kafka:<region>:<Account AWS-id>:cluster/msk/<cluster uuid>' ]
[ CATALOG_ROLE { 'SESSION' | 'catalog-role-arn-string' } ]
[ CREATE EXTERNAL DATABASE IF NOT EXISTS ]
[ CATALOG_ID 'Amazon Web Services account ID containing Glue or Lake Formation
 database' ]
```

La sintassi seguente descrive il comando CREATE EXTERNAL SCHEMA utilizzato per fare riferimento ai dati utilizzando una query federata su RDS POSTGRES o Aurora PostgreSQL. È inoltre possibile creare uno schema esterno che faccia riferimento a fonti di streaming, come Kinesis Data Streams. Per ulteriori informazioni, consulta Esecuzione di query su dati con query federate in Amazon Redshift.

```
CREATE EXTERNAL SCHEMA [IF NOT EXISTS] local_schema_name
FROM POSTGRES
DATABASE 'federated_database_name' [SCHEMA 'schema_name']
URI 'hostname' [ PORT port_number ]
```

```
IAM_ROLE { default | 'arn:aws:iam::<Account AWS-id>:role/<role-name>' }
SECRET_ARN 'ssm-secret-arn'
```

La sintassi seguente descrive il comando CREATE EXTERNAL SCHEMA utilizzato per fare riferimento ai dati utilizzando una query federata su RDS MySQL o Aurora MySQL. Per ulteriori informazioni, consulta Esecuzione di query su dati con query federate in Amazon Redshift.

```
CREATE EXTERNAL SCHEMA [IF NOT EXISTS] local_schema_name
FROM MYSQL
DATABASE 'federated_database_name'
URI 'hostname' [ PORT port_number ]
IAM_ROLE { default | 'arn:aws:iam::<Account AWS-id>:role/<role-name>' }
SECRET_ARN 'ssm-secret-arn'
```

La sintassi seguente descrive il comando CREATE EXTERNAL SCHEMA utilizzato per fare riferimento ai dati in un flusso Kinesis. Per ulteriori informazioni, consulta Importazione di dati in streaming.

```
CREATE EXTERNAL SCHEMA [IF NOT EXISTS] schema_name
FROM KINESIS
IAM_ROLE { default | 'arn:aws:iam::<Account AWS-id>:role/<role-name>' }
```

La sintassi seguente descrive il comando CREATE EXTERNAL SCHEMA utilizzato per fare riferimento al cluster Amazon Managed Streaming per Apache Kafka e i relativi argomenti da cui importare dati. CLUSTER_ARN specifica il cluster Amazon MSK da cui stai leggendo i dati. Per ulteriori informazioni, consulta Importazione di dati in streaming.

```
CREATE EXTERNAL SCHEMA [IF NOT EXISTS] schema_name
FROM MSK
IAM_ROLE { default | 'arn:aws:iam::<Account AWS-id>:role/<role-name>' }
AUTHENTICATION { none | iam }
CLUSTER_ARN 'msk-cluster-arn';
```

La sintassi seguente descrive il comando CREATE EXTERNAL SCHEMA utilizzato per fare riferimento ai dati utilizzando una query tra database.

```
CREATE EXTERNAL SCHEMA local_schema_name
FROM  REDSHIFT
DATABASE 'redshift_database_name' SCHEMA 'redshift_schema_name'
```

Parametri

IF NOT EXISTS

Clausola che indica che se lo schema specificato esiste già, il comando non deve apportare modifiche e deve restituire un messaggio che lo schema esiste, piuttosto che terminare con un errore. Questa clausola è utile durante lo scripting, quindi lo script non fallisce se CREATE EXTERNAL SCHEMA tenta di creare uno schema già esistente.

local_schema_name

Il nome del nuovo schema esterno. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

FROM [DATA CATALOG] | HIVE METASTORE | POSTGRES | MYSQL | KINESIS | MSK | REDSHIFT

Parola chiave che indica dove si trova il database esterno.

DATA CATALOG indica che il database esterno è definito nel catalogo dati di Athena o AWS Glue Data Catalog.

Se il database esterno è definito in un catalogo dati esterno in una regione AWS diversa, è necessario il parametro REGION. DATA CATALOG è il valore predefinito.

HIVE METASTORE indica che il database esterno è definito in un metastore Apache Hive. Se è specificato HIVE METASTORE, l'URI è obbligatorio.

POSTGRES indica che il database esterno è definito in RDS PostgreSQL o Aurora PostgreSQL.

MYSQL indica che il database esterno è definito in RDS MySQL o Aurora MySQL.

KINESIS indica che l'origine dati è un flusso di Kinesis Data Streams.

MSK indica che l'origine dati è un argomento di Amazon MSK.

FROM REDSHIFT

Una parola chiave che indica che il database si trova in Amazon Redshift.

DATABASE 'nome_database_redshift' SCHEMA 'nome_schema_redshift'

Il nome del database Amazon Redshift.

nome_schema redshift indica lo schema in Amazon Redshift. Il valore di timeout di default per nome_schema redshift è `public`.

DATABASE 'nome_database_federato'

Una parola chiave che indica il nome del database esterno in un motore del database PostgreSQL o MySQL supportato.

[SCHEMA 'nome_schema']

nome_schema indica lo schema in un motore del database PostgreSQL supportato. Lo schema_name predefinito è `public`.

Non è possibile specificare uno SCHEMA quando si imposta una query federata su un motore di database MySQL supportato.

REGION 'aws-region'

Se il database esterno è definito in un catalogo dati Athena o nella AWS Glue Data Catalog, la AWS regione in cui si trova il database. Questo parametro è obbligatorio se il database è definito in un catalogo dati esterno.

URI 'hive_metastore_uri' [PORT port_number]

L'URI del nome host e il numero di porta di un motore del database PostgreSQL o MySQL supportato. Il valore hostname è il nodo principale del set di repliche. L'endpoint deve essere raggiungibile (instradabile) dal cluster Amazon Redshift. Il port_number predefinito per PostgreSQL è 5432. Il port_number predefinito per MySQL è 3306.

Se il database si trova in un metastore Hive, specifica l'URI e facoltativamente il numero di porta per il metastore. Il numero di porta predefinito è 9083.

Un URI non contiene una specifica di protocollo ("http://"). Un URI valido esempio è `uri '172.10.10.10'`.

> **Note**
>
> Il motore del database PostgreSQL o MySQL supportato deve essere nello stesso VPC del cluster Amazon Redshift. Creare un gruppo di sicurezza che colleghi Amazon Redshift e RDS PostgreSQL o Aurora PostgreSQL.

IAM_ROLE { default | 'SESSION' | 'arn:aws:iam::*<Account AWS-id>*:role/*<role-name>*' }

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando CREATE EXTERNAL SCHEMA.

Utilizzare `'SESSION'` se ci si connette al cluster Amazon Redshift utilizzando un'identità federata e si accede alle tabelle dallo schema esterno creato con questo comando. Per ulteriori informazioni, consulta l'argomento relativo all'utilizzo di un'identità federata per gestire l'accesso di Amazon Redshift alle risorse locali e alle tabelle esterne di Amazon Redshift Spectrum, che illustra come configurare l'identità federata. Si noti che questa configurazione, che utilizza `'SESSION'` al posto dell'ARN, può essere utilizzata solo se lo schema è stato creato utilizzando `DATA CATALOG`.

L'Amazon Resource Name (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione. Come minimo, il ruolo IAM deve disporre dell'autorizzazione per eseguire un'operazione LIST sul bucket Amazon S3 a cui accedere e un'operazione GET sugli oggetti Amazon S3 contenuti nel bucket.

Quanto segue mostra la sintassi per la stringa di parametro IAM_ROLE per un singolo ARN.

```
IAM_ROLE 'arn:aws:iam::<aws-account-id>:role/<role-name>'
```

È possibile concatenare i ruoli in modo che il cluster possa presumere un altro ruolo IAM, possibilmente appartenente a un altro account. Puoi concatenare fino a 10 ruoli. Per un esempio di concatenamento di ruoli, vedi Concatenazione di ruoli IAM per Amazon Redshift Spectrum.

Per collegare a questo ruolo IAM una policy di autorizzazioni IAM simile alla seguente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessSecret",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
```

```
            "Resource": "arn:aws:secretsmanager:us-west-2:123456789012:secret:my-
rds-secret-VNenFy"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword",
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*"
        }
    ]
}
```

Per la procedura per creare un ruolo IAM da utilizzare con la query federata, consultare Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate.

> **Note**
>
> Non includere spazi nell'elenco dei ruoli concatenati.

Quanto segue mostra la sintassi per concatenare tre ruoli.

```
IAM_ROLE 'arn:aws:iam::<aws-account-id>:role/<role-1-name>,arn:aws:iam::<aws-
account-id>:role/<role-2-name>,arn:aws:iam::<aws-account-id>:role/<role-3-name>'
```

SECRET_ARN 'ssm-secret-arn'

L'Amazon Resource Name (ARN) di un segreto del motore di database PostgreSQL o MySQL supportato creato utilizzando. AWS Secrets Manager Per informazioni su come creare e recuperare un ARN per un segreto, consultare Creazione di un segreto di base e Recupero del valore segreto nella Guida per l'utente di AWS Secrets Manager .

CATALOG_ROLE { 'SESSION' | catalog-role-arn-string}

Utilizzare `'SESSION'` per la connessione al cluster Amazon Redshift utilizzando un'identità federata per l'autenticazione e l'autorizzazione al catalogo di dati. Per ulteriori informazioni sul completamento della procedura per l'identità federata, consulta l'argomento relativo all'utilizzo di un'identità federata per gestire l'accesso di Amazon Redshift alle risorse locali e alle tabelle

esterne di Amazon Redshift Spectrum. Si noti che il ruolo `'SESSION'` può essere utilizzato solo se lo schema viene creato in DATA CATALOG.

Il nome della risorsa Amazon (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione del catalogo dati.

Se CATALOG_ROLE non viene specificato, Amazon Redshift utilizza l'IAM_ROLE specificato. Il ruolo del catalogo deve disporre dell'autorizzazione per accedere al Data Catalog in AWS Glue o Athena. Per ulteriori informazioni, consulta Policy IAM per Amazon Redshift Spectrum.

Quanto segue mostra la sintassi per la stringa di parametro CATALOG_ROLE per un singolo ARN.

```
CATALOG_ROLE 'arn:aws:iam::<aws-account-id>:role/<catalog-role>'
```

È possibile concatenare i ruoli in modo che il cluster possa presumere un altro ruolo IAM, possibilmente appartenente a un altro account. Puoi concatenare fino a 10 ruoli. Per ulteriori informazioni, consulta Concatenazione di ruoli IAM per Amazon Redshift Spectrum.

> **Note**
>
> L'elenco di ruoli concatenati non deve includere spazi.

Quanto segue mostra la sintassi per concatenare tre ruoli.

```
CATALOG_ROLE 'arn:aws:iam::<aws-account-id>:role/<catalog-role-1-
name>,arn:aws:iam::<aws-account-id>:role/<catalog-role-2-name>,arn:aws:iam::<aws-
account-id>:role/<catalog-role-3-name>'
```

CREATE EXTERNAL DATABASE IF NOT EXISTS

Clausola che crea un database esterno con il nome specificato dall'argomento DATABASE, se il database esterno specificato non esiste. Se il database esterno specificato esiste, il comando non apporta modifiche. In questo caso, il comando restituisce un messaggio che il database esterno esiste, anziché terminare con un errore.

> **Note**
>
> CREATE EXTERNAL DATABASE IF NOT EXISTS non può essere usato con HIVE METASTORE.
> Per utilizzare CREATE EXTERNAL DATABASE IF NOT EXISTS con il catalogo dati abilitato per AWS Lake Formation, è necessaria l'autorizzazione `CREATE_DATABASE` sul catalogo dati.

CATALOG_ID "ID account Amazon Web Services contenente il database Glue o Lake Formation"

L'ID dell'account in cui è archiviato il database del catalogo dati.

`CATALOG_ID` può essere specificato solo se si prevede di connettersi al cluster Amazon Redshift o ad Amazon Redshift Serverless utilizzando un'identità federata per l'autenticazione e l'autorizzazione al catalogo di dati configurando una delle seguenti impostazioni:

- `CATALOG_ROLE` Da a `'SESSION'`
- `IAM_ROLE` su `'SESSION'` e `'CATALOG_ROLE'` impostati sul valore predefinito

Per ulteriori informazioni sul completamento della procedura per l'identità federata, consulta l'argomento relativo all'utilizzo di un'identità federata per gestire l'accesso di Amazon Redshift alle risorse locali e alle tabelle esterne di Amazon Redshift Spectrum.

AUTHENTICATION

Il tipo di autenticazione definito per l'importazione di dati in streaming. L'importazione di dati in streaming con i tipi di autenticazione funziona con Amazon Managed Streaming per Apache Kafka. I tipi di `AUTHENTICATION` sono i seguenti:

- none: specifica che non è presente alcuna procedura di autenticazione.
- iam: specifica l'autenticazione IAM. Quando scegli questa opzione, assicurati che il ruolo IAM disponga delle autorizzazioni per l'autenticazione IAM. Per ulteriori informazioni sulla definizione dello schema esterno, consulta Nozioni di base sull'importazione dati in streaming da Amazon Managed Streaming per Apache Kafka.

CLUSTER_ARN

Per l'importazione di dati in streaming, l'identificatore del cluster Amazon Managed Streaming per Apache Kafka da cui stai eseguendo lo streaming. Per ulteriori informazioni, consulta Importazione dati in streaming.

Note per l'utilizzo

Per i limiti per l'uso del catalogo di dati Athena, consulta Limiti di Athena in Riferimenti generali di AWS.

Per i limiti relativi all'utilizzo di AWS Glue Data Catalog, consulta AWS Glue Limiti in. Riferimenti generali di AWS

Questi limiti non si applicano a un metastore Hive.

È presente un massimo di 9900 schemi per database. Per ulteriori informazioni, consulta Quote e limiti nella Guida alla gestione di Amazon Redshift.

Per annullare la registrazione dello schema, utilizzare il comando DROP SCHEMA.

Per visualizzare i dettagli degli schemi esterni, eseguire una query sulle viste di sistema seguenti:

- SVV_EXTERNAL_SCHEMAS
- SVV_EXTERNAL_TABLES
- SVV_EXTERNAL_COLUMNS

Esempi

L'esempio seguente crea uno schema esterno che utilizza un database in un catalogo dati denominato `sampledb` nella regione Stati Uniti occidentali (Oregon). Usa questo esempio con un Athena o un catalogo di AWS Glue dati.

```
create external schema spectrum_schema
from data catalog
database 'sampledb'
region 'us-west-2'
iam_role 'arn:aws:iam::123456789012:role/MySpectrumRole';
```

L'esempio seguente crea uno schema esterno e un nuovo database esterno denominato `spectrum_db`.

```
create external schema spectrum_schema
from data catalog
database 'spectrum_db'
```

```
iam_role 'arn:aws:iam::123456789012:role/MySpectrumRole'
create external database if not exists;
```

L'esempio seguente crea uno schema esterno che utilizza un database metastore Hive denominato `hive_db`.

```
create external schema hive_schema
from hive metastore
database 'hive_db'
uri '172.10.10.10' port 99
iam_role 'arn:aws:iam::123456789012:role/MySpectrumRole';
```

Nell'esempio seguente sono concatenati i ruoli per utilizzare il ruolo `myS3Role` per accedere ad Amazon S3 ed è utilizzato `myAthenaRole` per l'accesso al catalogo dati. Per ulteriori informazioni, consulta Concatenazione di ruoli IAM per Amazon Redshift Spectrum.

```
create external schema spectrum_schema
from data catalog
database 'spectrum_db'
iam_role 'arn:aws:iam::123456789012:role/myRedshiftRole,arn:aws:iam::123456789012:role/
myS3Role'
catalog_role 'arn:aws:iam::123456789012:role/myAthenaRole'
create external database if not exists;
```

Nell'esempio seguente viene creato uno schema esterno che fa riferimento a un database Aurora PostgreSQL.

```
CREATE EXTERNAL SCHEMA [IF NOT EXISTS] myRedshiftSchema
FROM POSTGRES
DATABASE 'my_aurora_db' SCHEMA 'my_aurora_schema'
URI 'endpoint to aurora hostname' PORT 5432
IAM_ROLE 'arn:aws:iam::123456789012:role/MyAuroraRole'
SECRET_ARN 'arn:aws:secretsmanager:us-east-2:123456789012:secret:development/
MyTestDatabase-AbCdEf'
```

Nell'esempio seguente viene creato uno schema esterno per fare riferimento al database sales_db importato nel cluster di consumer.

```
CREATE EXTERNAL SCHEMA sales_schema FROM REDSHIFT DATABASE 'sales_db' SCHEMA 'public';
```

Nell'esempio seguente viene creato uno schema esterno che fa riferimento a un database Aurora MySQL.

```
CREATE EXTERNAL SCHEMA [IF NOT EXISTS] myRedshiftSchema
FROM MYSQL
DATABASE 'my_aurora_db'
URI 'endpoint to aurora hostname'
IAM_ROLE 'arn:aws:iam::123456789012:role/MyAuroraRole'
SECRET_ARN 'arn:aws:secretsmanager:us-east-2:123456789012:secret:development/
MyTestDatabase-AbCdEf'
```

CREATE EXTERNAL TABLE

Crea una nuova tabella esterna nello schema specificato. Tutte le tabelle esterne devono essere create in uno schema esterno. Il percorso di ricerca non è supportato per schemi esterni e tabelle esterne. Per ulteriori informazioni, consulta CREATE EXTERNAL SCHEMA.

Oltre alle tabelle esterne create utilizzando il comando CREATE EXTERNAL TABLE, Amazon Redshift può fare riferimento a tabelle esterne definite in un AWS Lake Formation catalogo AWS Glue or o in un metastore Apache Hive. Utilizzare il comando CREATE EXTERNAL SCHEMA per registrare un database esterno definito nel catalogo esterno e rendere disponibili le tabelle esterne per l'uso in Amazon Redshift. Se la tabella esterna esiste in un AWS Lake Formation catalogo AWS Glue o in un metastore Hive, non è necessario creare la tabella utilizzando CREATE EXTERNAL TABLE. Per visualizzare le tabelle esterne, eseguire una query sulla vista di sistema SVV_EXTERNAL_TABLES.

Eseguendo il comando CREATE EXTERNAL TABLE AS, è possibile creare una tabella esterna basata sulla definizione di colonna da una query e scrivere i risultati di tale query in Amazon S3. I risultati sono in formato Apache Parquet o testo delimitato. Se la tabella esterna dispone di una o più chiavi di partizione, Amazon Redshift partiziona i nuovi file in base a tali chiavi di partizione e registra automaticamente le nuove partizioni nel catalogo esterno. Per ulteriori informazioni su CREATE EXTERNAL TABLE AS, consultare Note per l'utilizzo.

È possibile eseguire query su una tabella esterna utilizzando la stessa sintassi SELECT utilizzata con altre tabelle Amazon Redshift. È inoltre possibile utilizzare la sintassi INSERT per scrivere nuovi file nella posizione della tabella esterna su Amazon S3. Per ulteriori informazioni, consulta INSERT (tabella esterna).

Per creare una vista con una tabella esterna, includi la clausola WITH NO SCHEMA BINDING nell'istruzione CREATE VIEW.

Non è possibile eseguire CREATE EXTERNAL TABLE all'interno di una transazione (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Privilegi richiesti

Per creare tabelle esterne, devi essere il proprietario dello schema esterno o un utente con privilegi avanzati. Per trasferire la proprietà di uno schema esterno, utilizza ALTER SCHEMA per cambiare il proprietario. L'accesso alle tabelle esterne è controllato dall'accesso allo schema esterno. Non puoi GRANT o REVOKE autorizzazioni per una tabella esterna. Puoi invece concedere o revocare USAGE sullo schema esterno.

Note per l'utilizzo dispone di informazioni aggiuntive sulle autorizzazioni specifiche per le tabelle esterne.

Sintassi

```
CREATE EXTERNAL TABLE
external_schema.table_name
(column_name data_type [, …] )
[ PARTITIONED BY (col_name data_type [, … ] )]
[ { ROW FORMAT DELIMITED row_format |
  ROW FORMAT SERDE 'serde_name'
  [ WITH SERDEPROPERTIES ( 'property_name' = 'property_value' [, ...] ) ] } ]
STORED AS file_format
LOCATION { 's3://bucket/folder/' | 's3://bucket/manifest_file' }
[ TABLE PROPERTIES ( 'property_name'='property_value' [, ...] ) ]
```

Di seguito è riportata la sintassi per CREATE EXTERNAL TABLE AS.

```
CREATE EXTERNAL TABLE
external_schema.table_name
[ PARTITIONED BY (col_name [, … ] ) ]
[ ROW FORMAT DELIMITED row_format ]
STORED AS file_format
LOCATION { 's3://bucket/folder/' }
[ TABLE PROPERTIES ( 'property_name'='property_value' [, ...] ) ]
 AS
 { select_statement }
```

Parametri

external_schema.table_name

Nome della tabella da creare, qualificato da un nome di schema esterno. Le tabelle esterne devono essere create in uno schema esterno. Per ulteriori informazioni, consulta CREATE EXTERNAL SCHEMA.

La lunghezza massima per il nome della tabella è 127 byte; i nomi più lunghi vengono troncati a 127 byte. Puoi utilizzare caratteri multibyte UTF-8 fino a un massimo di quattro byte. Amazon Redshift applica un limite di 9.900 tabelle per cluster, comprese le tabelle temporanee definite dall'utente e le tabelle temporanee create da Amazon Redshift durante l'elaborazione delle query o la manutenzione del sistema. Facoltativamente, puoi qualificare il nome della tabella con il nome del database. Nell'esempio seguente, il nome del database è `spectrum_db`, il nome dello schema esterno è `spectrum_schema` e il nome della tabella è `test`.

```
create external table spectrum_db.spectrum_schema.test (c1 int)
stored as parquet
location 's3://mybucket/myfolder/';
```

Se il database o lo schema specificato non esiste, la tabella non viene creata e l'istruzione restituisce un errore. Non è possibile creare tabelle o viste nei database di sistema `template0`, `template1`, `padb_harvest` o `sys:internal`.

Il nome della tabella deve essere un nome univoco per lo schema specificato.

Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

(column_name data_type)

Il nome e il tipo di dati di ciascuna colonna creata.

La lunghezza massima per il nome della colonna è 127 byte; i nomi più lunghi vengono troncati a 127 byte. Puoi utilizzare caratteri multibyte UTF-8 fino a un massimo di quattro byte. Non puoi specificare i nomi di colonna `"$path"` o `"$size"`. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

Per impostazione predefinita, Amazon Redshift crea tabelle esterne con le pseudocolonne `$path` e `$size`. Puoi disabilitare la creazione di pseudocolonne per una sessione impostando il parametro di configurazione `spectrum_enable_pseudo_columns` su `false`. Per ulteriori informazioni, consulta Pseudocolonne .

Se le pseudocolonne sono abilitate, il numero massimo di colonne che è possibile definire in una singola tabella è 1.598. Se le pseudocolonne non sono abilitate, il numero massimo di colonne che puoi definire in una singola tabella è 1.600.

Se stai creando una "tabella di grandi dimensioni", assicurati che il tuo elenco di colonne non superi i limiti della larghezza delle righe per i risultati intermedi durante i carichi e l'elaborazione delle query. Per ulteriori informazioni, consulta Note per l'utilizzo.

Per un comando CREATE EXTERNAL TABLE AS., non è necessario un elenco di colonne, poiché queste derivano dalla query.

data_type

Sono supportate le seguenti Tipi di dati:

- SMALLINT (INT2)
- INTEGER (INT, INT4)
- BIGINT (INT8)
- DECIMAL (NUMERIC)
- REAL (FLOAT4)
- DOUBLE PRECISION (FLOAT8)
- BOOLEAN (BOOL)
- CHAR (CHARACTER)
- VARCHAR (CHARACTER VARYING)
- VARBYTE (CHARACTER VARYING): può essere utilizzato con file di dati Parquet e ORC e solo con colonne non partizionate.
- DATE: può essere utilizzato solo con i file di testo, Parquet o file di dati ORC oppure come colonna di partizione.
- TIMESTAMP

Per DATE, è possibile utilizzare i formati come descritto di seguito. Per i valori mensili rappresentati tramite cifre, sono supportati i seguenti formati:

- `mm-dd-yyyy`, ad esempio, `05-01-2017`. Questa è l'impostazione predefinita.
- `yyyy-mm-dd`, dove l'anno è rappresentato da più di 2 cifre. Ad esempio, `2017-05-01`.

Per i valori mensili rappresentati tramite l'abbreviazione di tre lettere, sono supportati i seguenti formati:

- `mmm-dd-yyyy`, ad esempio, `may-01-2017`. Questa è l'impostazione predefinita.
- `dd-mmm-yyyy`, dove l'anno è rappresentato da più di 2 cifre. Ad esempio, `01-may-2017`.
- `yyyy-mmm-dd`, dove l'anno è rappresentato da più di 2 cifre. Ad esempio, `2017-may-01`.

Per i valori dell'anno che sono costantemente inferiori a 100, l'anno viene calcolato nel modo seguente:

- Se l'anno è inferiore a 70, l'anno viene calcolato come anno più 2000. Ad esempio, la data 05-01-17 nel formato `mm-dd-yyyy` viene convertito in `05-01-2017`.
- Se l'anno è inferiore a 100 ma superiore a 69, l'anno viene calcolato come anno più 1900. Ad esempio, la data 05-01-89 nel formato `mm-dd-yyyy` viene convertito in `05-01-1989`.
- Per i valori dell'anno rappresentati da due cifre, aggiungi zeri iniziali per rappresentare l'anno in 4 cifre.

I valori di timestamp nei file di testo devono essere nel formato `yyyy-mm-dd HH:mm:ss.SSSSSS`, come il seguente valore di timestamp `2017-05-01 11:30:59.000000`.

La lunghezza di una colonna VARCHAR è definita in termini di byte, non caratteri. Ad esempio, una colonna VARCHAR(12) può contenere 12 caratteri a byte singolo o 6 caratteri a due byte. Quando invii una query a una tabella esterna, i risultati sono troncati in modo da essere compatibili con le dimensioni definite della colonna senza restituire un errore. Per ulteriori informazioni, consulta Storage e intervalli.

Per garantire le prestazioni migliori, consigliamo di specificare le dimensioni della colonna più piccole compatibili con i dati. Per individuare le dimensioni massime in byte per i valori di una colonna, utilizza la funzione OCTET_LENGTH. Nell'esempio seguente viene restituita la dimensione massima dei valori nella colonna dell'e-mail.

```
select max(octet_length(email)) from users;

max
---
 62
```

PARTITIONED BY (col_name data_type [, …])

Una clausola che definisce una tabella partizionata con una o più colonne di partizione. Viene utilizzata una directory di dati separata per ciascuna combinazione specificata, che può migliorare le prestazioni della query in alcune circostanze. Le colonne partizionate non esistono all'interno

dei dati della tabella stessa. Se usi un valore per col_name uguale a una colonna della tabella, viene restituito un errore.

Dopo aver creato una tabella partizionata, modificare la tabella utilizzando un'istruzione ALTER TABLE ... ADD PARTITION per registrare nuove partizioni nel catalogo esterno. Quando si aggiunge una partizione, si definisce la posizione della sottocartella su Amazon S3 contenenti i dati della partizione.

Ad esempio, se la tabella `spectrum.lineitem_part` è definita con `PARTITIONED BY (l_shipdate date)`, esegui il seguente comando ALTER TABLE per aggiungere una partizione.

```
ALTER TABLE spectrum.lineitem_part ADD PARTITION (l_shipdate='1992-01-29')
LOCATION 's3://spectrum-public/lineitem_partition/l_shipdate=1992-01-29';
```

Se utilizzi CREATE EXTERNAL TABLE AS, non è necessario eseguire ALTER TABLE…ADD PARTITION. Amazon Redshift registra automaticamente le nuove partizioni nel catalogo esterno. Amazon Redshift scrive inoltre automaticamente i dati corrispondenti nelle partizioni Amazon S3 in base alla chiave o alle chiavi di partizione definite nella tabella.

Per visualizzare le partizioni, eseguire una query sulla vista di sistema SVV_EXTERNAL_PARTITIONS.

> **Note**
>
> Per un comando CREATE EXTERNAL TABLE AS, non è necessario specificare il tipo di dati della colonna di partizione perché questa colonna è derivata dalla query.

ROW FORMAT DELIMITED rowformat

Clausola che specifica il formato dei dati sottostanti. I valori possibili di rowformat sono indicati di seguito:

- LINES TERMINATED BY 'delimiter'
- FIELDS TERMINATED BY 'delimiter'

Specifica un singolo carattere ASCII per 'delimiter'. È possibile specificare caratteri ASCII non stampabili utilizzando un ottale, nel formato `'\`*ddd*`'` dove *d* è una cifra ottale (0-7) fino a '\177'. L'esempio seguente specifica il carattere BEL (bell) usando l'ottale.

```
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\007'
```

Se ROW FORMAT è omesso, il formato predefinito è DELIMITED FIELDS TERMINATED BY '\A' (inizio dell'intestazione) e LINES TERMINATED BY '\n' (newline).

ROW FORMAT SERDE 'serde_name' , [WITH SERDEPROPERTIES ('property_name' = 'property_value' [, ...])]

Clausola che specifica il formato SERDE dei dati sottostanti.

'serde_name'

Nome della SerDe. È possibile specificare utilizzando i seguenti formati.

- org.apache.hadoop.hive.serde2. RegexSerDe
- com.amazonaws.glue.serde. GrokSerDe
- org.apache.hadoop.hive.serde2.OpenCSVSerde

  Questo parametro supporta la seguente SerDe proprietà per OpenCsvSerde:

  ```
  'wholeFile' = 'true'
  ```

  Impostare la proprietà `wholeFile`a proprietà `true` per analizzare correttamente i nuovi caratteri di riga (\ n) all'interno di stringhe virgolate per le richieste OpenCSV.
- org.openx.data.jsonserde. JsonSerDe
  - Il JSON SERDE supporta anche i file Ion.
  - Il JSON deve essere ben formato.
  - I timestamp in Ion e JSON devono utilizzare il formato ISO8601.
  - Questo parametro supporta la seguente SerDe proprietà per JsonSerDe:

    ```
    'strip.outer.array'='true'
    ```

    Elabora i file Ion/JSON che contengono un array molto grande racchiuso tra parentesi esterne ([…]) come se contenesse diversi record JSON all'interno dell'array.
- com.amazon.ionhiveserde. IonHiveSerDe

  Il formato Amazon ION fornisce formati di testo e binari, oltre ai tipi di dati. Per una tabella esterna che fa riferimento ai dati in formato ION, devi mappare ogni colonna della tabella

verso l'elemento corrispondente nei dati di formato ION. Per ulteriori dettagli, consultare Amazon Ion. È inoltre necessario specificare i formati di input e di output.

WITH SERDEPROPERTIES ('property_name' = 'property_value' [, ...])]

Facoltativamente, specifica i nomi e i valori delle proprietà, separati da virgole.

Se ROW FORMAT è omesso, il formato predefinito è DELIMITED FIELDS TERMINATED BY '\A' (inizio dell'intestazione) e LINES TERMINATED BY '\n' (newline).

STORED AS file_format

Il formato per i file di dati.

I formati validi sono:

- PARQUET
- RCFILE (ColumnarSerDe solo per uso di dati, non) LazyBinaryColumnarSerDe
- SEQUENCEFILE
- TEXTFILE (per file di testo, inclusi i file JSON)
- ORC
- AVRO
- INPUTFORMAT 'input_format_classname' OUTPUTFORMAT 'output_format_classname'

Il comando CREATE EXTERNAL TABLE AS supporta solo due formati di file, TEXTFILE e PARQUET.

Per INPUTFORMAT e OUTPUTFORMAT, specifica un nome di classe, come illustrato nel seguente esempio.

```
'org.apache.hadoop.mapred.TextInputFormat'
```

LOCATION { 's3://bucket/folder/' | 's3://bucket/manifest_file'}

Il percorso della cartella o del bucket Amazon S3 con i file di dati o un file manifest che contiene un elenco di percorsi di oggetti Amazon S3. I bucket devono trovarsi nella stessa AWS regione del cluster Amazon Redshift. Per un elenco delle AWS regioni supportate, consulta. Considerazioni su Amazon Redshift Spectrum

Se il percorso specifica un bucket o una cartella, ad esempio `'s3://mybucket/custdata/'`, Redshift Spectrum esegue la scansione dei file nel bucket o nella cartella specificati e in qualsiasi

sottocartella. Redshift Spectrum ignora i file e i file nascosti che iniziano con un punto o un trattino basso.

Se il percorso specifica un file manifest, l'argomento `'s3://bucket/manifest_file'` deve fare riferimento esplicitamente a un singolo file, ad esempio `'s3://mybucket/manifest.txt'`. Non può fare riferimento a un prefisso della chiave.

Il manifest è un file di testo in formato JSON che elenca l'URL di ciascun file che deve essere caricato da Amazon S3 e la dimensione del file, in byte. L'URL include il nome del bucket e il percorso completo dell'oggetto per il file. I file specificati nel manifesto possono trovarsi in diversi bucket, ma tutti i bucket devono trovarsi nella stessa AWS regione del cluster Amazon Redshift. Se un file viene elencato due volte, viene caricato due volte. L'esempio seguente mostra il JSON per un manifest che carica tre file.

```
{
  "entries": [
    {"url":"s3://mybucket-alpha/custdata.1", "meta": { "content_length":
 5956875 } },
    {"url":"s3://mybucket-alpha/custdata.2", "meta": { "content_length":
 5997091 } },
    {"url":"s3://mybucket-beta/custdata.1", "meta": { "content_length": 5978675 } }
  ]
}
```

È possibile rendere obbligatoria l'inclusione di un particolare file. A tale scopo, includere un'opzione `mandatory` a livello di file nel manifest. Quando si esegue una query su una tabella esterna con un file obbligatorio mancante, l'istruzione SELECT ha esito negativo. Assicurarsi che tutti i file inclusi nella definizione della tabella esterna siano presenti. Se non sono tutti presenti, viene visualizzato un errore che mostra il primo file obbligatorio che non viene trovato. L'esempio seguente mostra il JSON per un manifest con l' opzione `mandatory` impostata su `true`.

```
{
  "entries": [
    {"url":"s3://mybucket-alpha/custdata.1", "mandatory":true, "meta":
 { "content_length": 5956875 } },
    {"url":"s3://mybucket-alpha/custdata.2", "mandatory":false, "meta":
 { "content_length": 5997091 } },
    {"url":"s3://mybucket-beta/custdata.1", "meta": { "content_length": 5978675 } }
  ]
}
```

Per fare riferimento ai file creati utilizzando UNLOAD, puoi utilizzare il manifest creato utilizzando UNLOAD con il parametro MANIFEST. Il file manifest è compatibile con un file manifest per COPY da Amazon S3, ma usa chiavi diverse. Le chiavi non utilizzate vengono ignorate.

TABLE PROPERTIES ('property_name'='property_value' [, ...])

Clausola che imposta la definizione della tabella delle proprietà della tabella.

> **Note**
>
> Le proprietà della tabella rispettano la distinzione tra maiuscole e minuscole.

'compression_type'='value'

Proprietà che imposta il tipo di compressione da utilizzare se il nome file non contiene un'estensione. Se imposti questa proprietà e c'è un'estensione di file, l'estensione viene ignorata e viene utilizzato il valore impostato dalla proprietà. I valori validi per il tipo di compressione sono i seguenti:

- bzip2
- gzip
- nessuno
- snappy

'data_cleansing_enabled'='true / false'

Questa proprietà imposta se la gestione dei dati è attiva per la tabella. Se 'data_cleansing_enabled' è impostata su true, la gestione dei dati è attiva per la tabella. Se 'data_cleansing_enabled' è impostata su false, la gestione dei dati non è attiva per la tabella. Di seguito è riportato l'elenco delle proprietà di gestione dei dati a livello di tabella controllate da questa proprietà:

- column_count_mismatch_handling
- invalid_char_handling
- numeric_overflow_handling
- replacement_char
- surplus_char_handling

Per alcuni esempi, consulta Esempi di gestione dei dati.

'invalid_char_handling'='valore'

Specifica l'azione da eseguire quando i risultati della query contengono valori di caratteri UTF-8 non validi. È possibile specificare le seguenti operazioni:

DISABILITATO

Non esegue la gestione dei caratteri non validi.

FAIL

Annulla le query che restituiscono dati contenenti valori UTF-8 non validi.

SET_TO_NULL

Sostituisce valori UTF-8 non validi con null.

DROP_ROW

Sostituisce ogni valore della riga con null.

REPLACE

Sostituisce il carattere non valido con il carattere sostitutivo specificato tramite `replacement_char`.

'replacement_char'='carattere'

Specifica il carattere sostitutivo da utilizzare quando si imposta `invalid_char_handling` su `REPLACE`.

'numeric_overflow_handling'='value'

Specifica l'azione da eseguire quando i dati ORC contengono un numero intero (ad esempio BIGINT o int64) maggiore della definizione di colonna (ad esempio SMALLINT o int16). È possibile specificare le seguenti operazioni:

DISABILITATO

La gestione dei caratteri non validi è disattivata.

FAIL

Se i dati includono caratteri non validi annullare la query.

SET_TO_NULL

Impostare i caratteri non validi su null.

DROP_ROW

Impostare ogni valore della riga su null.

'surplus_bytes_handling'='value'

Specifica come gestire i dati caricati che superano la lunghezza del tipo di dati definito per le colonne contenenti dati VARBYTE. Per impostazione predefinita, Redshift Spectrum imposta il valore su null per i dati che superano la larghezza della colonna.

È possibile specificare le seguenti operazioni da eseguire quando la query restituisce dati che superano la larghezza del tipo di dati:

SET_TO_NULL

Sostituisce i dati che superano la larghezza della colonna con null.

DISABILITATO

Non esegue la gestione dei byte in eccedenza.

FAIL

Annulla le query che restituiscono dati che superano la larghezza della colonna.

DROP_ROW

Elimina tutte le righe contenenti dati che superano la larghezza della colonna.

TRUNCATE

Rimuove i caratteri che superano il numero massimo di caratteri definito per la colonna.

'surplus_char_handling'='valore'

Specifica come gestire i dati caricati che superano la lunghezza del tipo di dati definito per le colonne contenenti dati VARCHAR, CHAR o stringa. Per impostazione predefinita, Redshift Spectrum imposta il valore su null per i dati che superano la larghezza della colonna.

È possibile specificare le seguenti operazioni da eseguire quando la query restituisce dati che superano la larghezza della colonna:

SET_TO_NULL

Sostituisce i dati che superano la larghezza della colonna con null.

DISABILITATO

Non esegue la gestione dei caratteri in eccedenza.

FAIL

Annulla le query che restituiscono dati che superano la larghezza della colonna.

DROP_ROW

Sostituisce ogni valore della riga con null.

TRUNCATE

Rimuove i caratteri che superano il numero massimo di caratteri definito per la colonna.

'column_count_mismatch_handling'='valore’

Determina se il file contiene più o meno valori in una riga rispetto al numero di colonne specificato nella definizione della tabella esterna. Questa proprietà è disponibile solo per il formato di file di testo non compresso. È possibile specificare le seguenti operazioni:

DISABILITATO

La gestione della mancata corrispondenza del conteggio di colonne è disattivata.

FAIL

La query ha esito negativo se viene rilevata una mancata corrispondenza del conteggio delle colonne.

SET_TO_NULL

Compila i valori mancanti con NULL e ignora i valori aggiuntivi in ogni riga.

DROP_ROW

Elimina tutte le righe che contengono errori relativi alla mancata corrispondenza del conteggio delle colonne dalla scansione.

'numRows'='row_count'

Proprietà che imposta il valore numRows per la definizione della tabella. Per aggiornare in modo esplicito le statistiche di una tabella esterna, imposta la proprietà numRows in modo da indicare le dimensioni della tabella. Amazon Redshift non analizza le tabelle esterne per generare le statistiche delle tabelle che l'ottimizzatore di query utilizza per generare un piano di query. Se le statistiche delle tabelle non sono impostate per una tabella esterna, Amazon

Redshift genera un piano di esecuzione delle query basato sul presupposto che le tabelle esterne sono le tabelle più grandi e quelle locali le più piccole.

'skip.header.line.count'='line_count'

Proprietà che imposta il numero di righe da saltare all'inizio di ogni file sorgente.

'serialization.null.format'=' '

Proprietà che specifica che Spectrum deve restituire un valore NULL quando esiste una corrispondenza esatta con il testo fornito in un campo.

'orc.schema.resolution'='mapping_type'

Una proprietà che imposta il tipo di mappatura della colonna per le tabelle che utilizzano il formato di dati ORC. La proprietà viene ignorata per tutti gli altri formati di dati.

I valori validi per il tipo di mappatura della colonna sono i seguenti:

- nome
- position

Se la proprietà orc.schema.resolution viene omessa, le colonne vengono mappate in base al nome per impostazione predefinita. Se orc.schema.resolution è impostata su un valore diverso da 'name' o 'position', le colonne vengono mappate in base alla posizione. Per ulteriori informazioni sulla mappatura delle colonne, consulta Mappatura delle colonne di una tabella esterna alle colonne ORC.

> **Note**
>
> Il comando COPY mappa sui file di dati ORC soltanto in base alla posizione. La proprietà della tabella orc.schema.resolution non ha effetto sul comportamento del comando COPY.

'write.parallel'='on / off'

Proprietà che imposta se CREATE EXTERNAL TABLE AS deve scrivere dati in parallelo. Per impostazione predefinita, CREATE EXTERNAL TABLE AS scrive i dati in parallelo in più file, in base al numero di sezioni nel cluster. L'opzione predefinita è "attiva". Quando 'write.parallel' è impostato su off, CREATE EXTERNAL TABLE AS scrive su uno o più file di dati in serie su Amazon S3. Questa proprietà della tabella si applica anche a qualsiasi istruzione INSERT successiva nella stessa tabella esterna.

‘write.maxfilesize.mb'=‘size'

Una proprietà che imposta la dimensione massima (in MB) di ogni file scritto su Amazon S3 da CREATE EXTERNAL TABLE AS. La dimensione deve essere un numero intero valido compreso tra 5 e 6200. La dimensione massima predefinita del file è 6.200 MB. Questa proprietà della tabella si applica anche a qualsiasi istruzione INSERT successiva nella stessa tabella esterna.

'write.kms.key.id'='valore'

Puoi specificare una AWS Key Management Service chiave per abilitare la crittografia lato server (SSE) per oggetti Amazon S3, dove il valore è uno dei seguenti:

- `auto`per utilizzare la AWS KMS chiave predefinita memorizzata nel bucket Amazon S3.
- kms-key che si specifica per crittografare i dati.

select_statement

Istruzione che inserisce una o più righe nella tabella esterna definendo qualsiasi query. Tutte le righe prodotte dalla query vengono scritte in Amazon S3 in formato testo o Parquet in base alla definizione della tabella.

Esempi

Una raccolta di esempi è disponibile all'indirizzo Esempi.

Note per l'utilizzo

Questo argomento contiene note di utilizzo per CREATE EXTERNAL TABLE. Non è possibile visualizzare i dettagli per le tabelle Amazon Redshift Spectrum che utilizzano le stesse risorse utilizzate per le tabelle Amazon Redshift standard come PG_TABLE_DEF, STV_TBL_PERM, PG_CLASS o information_schema. Se il tuo strumento di business intelligence o di analisi non riconosce le tabelle esterne Redshift Spectrum, configura l'applicazione per eseguire la query su SVV_EXTERNAL_TABLES e SVV_EXTERNAL_COLUMNS.

CREATE EXTERNAL TABLE AS

In alcuni casi, puoi eseguire il comando CREATE EXTERNAL TABLE AS su un catalogo AWS Glue dati, un catalogo AWS Lake Formation esterno o un metastore Apache Hive. In questi casi, si utilizza un ruolo AWS Identity and Access Management (IAM) per creare lo schema esterno. Questo ruolo IAM deve disporre di autorizzazioni di lettura e scrittura per Amazon S3.

Se si utilizza un catalogo Lake Formation, il ruolo IAM deve disporre dell'autorizzazione per creare una tabella nel catalogo. In questo caso, deve anche disporre dell'autorizzazione per la posizione del data lake sul percorso di destinazione di Amazon S3. Questo ruolo IAM diventa il proprietario della nuova tabella AWS Lake Formation .

Per garantire che i nomi file siano univoci, Amazon Redshift utilizza il seguente formato per il nome di ogni file caricato in Amazon S3 per impostazione predefinita.

*<date>*_*<time>*_*<microseconds>*_*<query_id>*_*<slice-number>*_part_*<part-number>*.*<format>*.

Un esempio è `20200303_004509_810669_1007_0001_part_00.parquet`.

Quando si esegue il comando CREATE EXTERNAL TABLE AS, considerare quanto segue:

- La posizione di Amazon S3 deve essere vuota.
- Quando si utilizza la clausola STORED AS, Amazon Redshift supporta solo i formati PARQUET e TEXTFILE.
- Non è necessario definire un elenco di definizioni di colonna. I nomi delle colonne e i tipi di dati delle colonne della nuova tabella esterna derivano direttamente dalla query SELECT.
- Non è necessario definire il tipo di dati della colonna di partizione nella clausola PARTITIONED BY. Se si specifica una chiave di partizione, il nome di questa colonna deve essere presente nel risultato della query SELECT. Quando si dispone di più colonne di partizione, il loro ordine nella query SELECT non ha importanza. Per creare la tabella esterna, Amazon Redshift utilizza l'ordine definito nella clausola PARTITIONED BY.
- Amazon Redshift suddivide automaticamente i file di output in cartelle di partizione in base ai valori della chiave di partizione. Per impostazione predefinita, Amazon Redshift rimuove le colonne di partizione dai file di output.
- La clausola LINES TERMINATE BY 'delimiter' non è supportata.
- La clausola ROW FORMAT SERDE 'serde_name' non è supportata.
- L'utilizzo dei file manifest non è supportato. Pertanto, non è possibile definire la clausola LOCATION in un file manifest su Amazon S3.
- Amazon Redshift aggiorna automaticamente la proprietà della tabella 'numRows' alla fine del comando.
- La proprietà di tabella 'compression_type' accetta solo 'none' o 'snappy' per il formato di file PARQUET.

- Amazon Redshift non consente la clausola LIMIT nella query SELECT esterna. È possibile invece utilizzare una clausola LIMIT nidificata.
- È possibile utilizzare STL_UNLOAD_LOG per tenere traccia dei file scritti in Amazon S3 da ogni operazione CREATE EXTERNAL TABLE AS.

Autorizzazioni per creare ed eseguire query sulle tabelle esterne

Per creare tabelle esterne, assicurarsi di essere il proprietario dello schema esterno o un utente con privilegi avanzati. Per trasferire la proprietà di uno schema esterno, utilizza ALTER SCHEMA. L'esempio seguente cambia il proprietario dello schema `spectrum_schema` in `newowner`.

```
alter schema spectrum_schema owner to newowner;
```

Per eseguire una query di Redshift Spectrum, sono necessarie le seguenti autorizzazioni:

- Autorizzazione di utilizzare lo schema
- Autorizzazione di creare tabelle temporanee nel database corrente

L'esempio seguente concede l'autorizzazione all'utilizzo dello schema `spectrum_schema` al gruppo di utenti `spectrumusers`.

```
grant usage on schema spectrum_schema to group spectrumusers;
```

L'esempio seguente concede l'autorizzazione temporanea per il database `spectrumdb` al gruppo di utenti `spectrumusers`.

```
grant temp on database spectrumdb to group spectrumusers;
```

Pseudocolonne

Per impostazione predefinita, Amazon Redshift crea tabelle esterne con le pseudocolonne $path e $size. Selezionare queste colonne per visualizzare il percorso ai file di dati su Amazon S3 e le dimensioni dei file di dati per ogni riga restituita da una query. I nomi di colonna $path e $size devono essere delimitati da virgolette doppie. La clausola SELECT * non restituisce le pseudocolonne. Devi includere in modo esplicito i nomi delle colonne $path e $size nella tua query, come indicato nel seguente esempio.

```
select "$path", "$size"
from spectrum.sales_part
where saledate = '2008-12-01';
```

Puoi disabilitare la creazione di pseudocolonne per una sessione impostando il parametro di configurazione spectrum_enable_pseudo_columns su false.

> **Important**
>
> La selezione di $size o $path comporta dei costi in quanto Redshift Spectrum esegue la scansione dei file di dati su Amazon S3 per determinare la dimensione del set di risultati. Per ulteriori informazioni sui prezzi, consultare Prezzi di Amazon Redshift.

Impostazione delle opzioni di gestione dati

È possibile impostare i parametri della tabella per specificare la gestione dell'input per i dati da sottoporre a query in tabelle esterne, tra cui:

- I caratteri in eccedenza nelle colonne contenenti dati VARCHAR, CHAR e stringa. Per ulteriori informazioni, consultare la proprietà della tabella esterna `surplus_char_handling`.
- I caratteri non validi nelle colonne contenenti dati VARCHAR, CHAR e stringa. Per ulteriori informazioni, consultare la proprietà della tabella esterna `invalid_char_handling`.
- Carattere sostitutivo da utilizzare quando si specifica REPLACE per la proprietà della tabella esterna `invalid_char_handling`.
- Gestione dell'eccedenza cast nelle colonne contenenti dati interi e decimali. Per ulteriori informazioni, consultare la proprietà della tabella esterna `numeric_overflow_handling`.
- Surplus_bytes_handling per specificare la gestione dell'input per i byte in eccesso nelle colonne contenenti dati varbyte. Per ulteriori informazioni, consultare la proprietà della tabella esterna `surplus_bytes_handling`.

Esempi

Nell'esempio seguente viene creata una tabella denominata SALES nello schema esterno Amazon Redshift denominato `spectrum`. I dati sono in file di testo delimitati da tabulazioni. La clausola TABLE PROPERTIES imposta la proprietà numRows su 170.000 righe.

A seconda dell'identità utilizzata per eseguire CREATE EXTERNAL TABLE, è possibile che sia necessario configurare delle autorizzazioni IAM. Come best practice, consigliamo di collegare le policy di autorizzazioni a un ruolo IAM, che quindi viene assegnato a utenti e gruppi secondo le necessità. Per ulteriori informazioni, consulta Identity and access management in Amazon Redshift.

```
create external table spectrum.sales(
salesid integer,
listid integer,
sellerid integer,
buyerid integer,
eventid integer,
saledate date,
qtysold smallint,
pricepaid decimal(8,2),
commission decimal(8,2),
saletime timestamp)
row format delimited
fields terminated by '\t'
stored as textfile
location 's3://redshift-downloads/tickit/spectrum/sales/'
table properties ('numRows'='170000');
```

L'esempio seguente crea una tabella che utilizza JsonSerDe per fare riferimento ai dati in formato JSON.

```
create external table spectrum.cloudtrail_json (
event_version int,
event_id bigint,
event_time timestamp,
event_type varchar(10),
awsregion varchar(20),
event_name varchar(max),
event_source varchar(max),
requesttime timestamp,
useragent varchar(max),
recipientaccountid bigint)
row format serde 'org.openx.data.jsonserde.JsonSerDe'
with serdeproperties (
'dots.in.keys' = 'true',
'mapping.requesttime' = 'requesttimestamp'
) location 's3://mybucket/json/cloudtrail';
```

Nell'esempio CREATE EXTERNAL TABLE AS seguente viene creata una tabella esterna non partizionata. Quindi il risultato della query SELECT viene scritto come Apache Parquet nella posizione Amazon S3 di destinazione.

```
CREATE EXTERNAL TABLE spectrum.lineitem
STORED AS parquet
LOCATION 'S3://mybucket/cetas/lineitem/'
AS SELECT * FROM local_lineitem;
```

Nell'esempio seguente viene creata una tabella esterna partizionata e vengono incluse le colonne di partizione nella query SELECT.

```
CREATE EXTERNAL TABLE spectrum.partitioned_lineitem
PARTITIONED BY (l_shipdate, l_shipmode)
STORED AS parquet
LOCATION 'S3://mybucket/cetas/partitioned_lineitem/'
AS SELECT l_orderkey, l_shipmode, l_shipdate, l_partkey FROM local_table;
```

Per un elenco di database esistenti nel catalogo dati esterno, eseguire una query sulla vista di sistema SVV_EXTERNAL_DATABASES.

```
select eskind,databasename,esoptions from svv_external_databases order by databasename;
```

```
eskind | databasename | esoptions
-------+--------------
+-------------------------------------------------------------------------------
     1 | default      | {"REGION":"us-
west-2","IAM_ROLE":"arn:aws:iam::123456789012:role/mySpectrumRole"}
     1 | sampledb     | {"REGION":"us-
west-2","IAM_ROLE":"arn:aws:iam::123456789012:role/mySpectrumRole"}
     1 | spectrumdb   | {"REGION":"us-
west-2","IAM_ROLE":"arn:aws:iam::123456789012:role/mySpectrumRole"}
```

Per visualizzare i dettagli delle tabelle esterne, eseguire una query sulle viste di sistema SVV_EXTERNAL_TABLES e SVV_EXTERNAL_COLUMNS.

L'esempio seguente esegue una query sulla vista SVV_EXTERNAL_TABLES.

```
select schemaname, tablename, location from svv_external_tables;
```

```
schemaname | tablename            | location
-----------+----------------------
+-------------------------------------------------------
spectrum   | sales                | s3://redshift-downloads/tickit/spectrum/sales
spectrum   | sales_part           | s3://redshift-downloads/tickit/spectrum/
sales_partition
```

L'esempio seguente esegue una query sulla vista SVV_EXTERNAL_COLUMNS.

```
select * from svv_external_columns where schemaname like 'spectrum%' and tablename
 ='sales';
```

```
schemaname | tablename | columnname | external_type | columnnum | part_key
-----------+-----------+------------+---------------+-----------+---------
spectrum   | sales     | salesid    | int           |         1 |        0
spectrum   | sales     | listid     | int           |         2 |        0
spectrum   | sales     | sellerid   | int           |         3 |        0
spectrum   | sales     | buyerid    | int           |         4 |        0
spectrum   | sales     | eventid    | int           |         5 |        0
spectrum   | sales     | saledate   | date          |         6 |        0
spectrum   | sales     | qtysold    | smallint      |         7 |        0
spectrum   | sales     | pricepaid  | decimal(8,2)  |         8 |        0
spectrum   | sales     | commission | decimal(8,2)  |         9 |        0
spectrum   | sales     | saletime   | timestamp     |        10 |        0
```

Per visualizzare le partizioni della tabella, usa la seguente query.

```
select schemaname, tablename, values, location
from svv_external_partitions
where tablename = 'sales_part';
```

```
schemaname | tablename  | values         | location
-----------+------------+----------------
+-------------------------------------------------------------------------
spectrum   | sales_part | ["2008-01-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-01
spectrum   | sales_part | ["2008-02-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-02
spectrum   | sales_part | ["2008-03-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-03
```

```
spectrum   | sales_part | ["2008-04-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-04
spectrum   | sales_part | ["2008-05-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-05
spectrum   | sales_part | ["2008-06-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-06
spectrum   | sales_part | ["2008-07-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-07
spectrum   | sales_part | ["2008-08-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-08
spectrum   | sales_part | ["2008-09-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-09
spectrum   | sales_part | ["2008-10-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-10
spectrum   | sales_part | ["2008-11-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-11
spectrum   | sales_part | ["2008-12-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-12
```

L'esempio seguente restituisce la dimensione totale dei file di dati correlati per una tabella esterna.

```
select distinct "$path", "$size"
   from spectrum.sales_part;

 $path                                                                      | $size
----------------------------------------------------------------------------+-------
s3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-01/ |  1616
s3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-02/ |  1444
s3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-02/ |  1444
```

Esempi di partizionamento

Per creare una tabella esterna partizionata per data, esegui il comando seguente.

```
create external table spectrum.sales_part(
salesid integer,
listid integer,
sellerid integer,
buyerid integer,
eventid integer,
dateid smallint,
qtysold smallint,
pricepaid decimal(8,2),
```

```
commission decimal(8,2),
saletime timestamp)
partitioned by (saledate date)
row format delimited
fields terminated by '|'
stored as textfile
location 's3://redshift-downloads/tickit/spectrum/sales_partition/'
table properties ('numRows'='170000');
```

Per aggiungere le partizioni, esegui il comando ALTER TABLE.

```
alter table spectrum.sales_part
add if not exists partition (saledate='2008-01-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-01/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-02-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-02/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-03-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-03/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-04-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-04/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-05-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-05/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-06-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-06/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-07-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-07/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-08-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-08/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-09-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-09/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-10-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-10/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-11-01')
```

```
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-11/';
alter table spectrum.sales_part
add if not exists partition (saledate='2008-12-01')
location 's3://redshift-downloads/tickit/spectrum/sales_partition/saledate=2008-12/';
```

Per selezionare i dati dalla tabella partizionata, esegui la seguente query.

```
select top 10 spectrum.sales_part.eventid, sum(spectrum.sales_part.pricepaid)
from spectrum.sales_part, event
where spectrum.sales_part.eventid = event.eventid
  and spectrum.sales_part.pricepaid > 30
  and saledate = '2008-12-01'
group by spectrum.sales_part.eventid
order by 2 desc;
```

```
eventid | sum
--------+---------
    914 | 36173.00
   5478 | 27303.00
   5061 | 26383.00
   4406 | 26252.00
   5324 | 24015.00
   1829 | 23911.00
   3601 | 23616.00
   3665 | 23214.00
   6069 | 22869.00
   5638 | 22551.00
```

Per visualizzare le partizioni delle tabelle, eseguire una query sulla vista di sistema SVV_EXTERNAL_PARTITIONS.

```
select schemaname, tablename, values, location from svv_external_partitions
where tablename = 'sales_part';
```

```
schemaname | tablename  | values         | location
-----------+------------+----------------
+-------------------------------------------------
spectrum   | sales_part | ["2008-01-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-01
spectrum   | sales_part | ["2008-02-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-02
```

```
spectrum   | sales_part | ["2008-03-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-03
spectrum   | sales_part | ["2008-04-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-04
spectrum   | sales_part | ["2008-05-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-05
spectrum   | sales_part | ["2008-06-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-06
spectrum   | sales_part | ["2008-07-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-07
spectrum   | sales_part | ["2008-08-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-08
spectrum   | sales_part | ["2008-09-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-09
spectrum   | sales_part | ["2008-10-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-10
spectrum   | sales_part | ["2008-11-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-11
spectrum   | sales_part | ["2008-12-01"] | s3://redshift-downloads/tickit/spectrum/
sales_partition/saledate=2008-12
```

Esempi di formati di riga

Di seguito viene mostrato un esempio di specifica dei parametri ROW FORMAT SERDE per i file di dati memorizzati nel formato AVRO.

```
create external table spectrum.sales(salesid int, listid int, sellerid int,
 buyerid int, eventid int, dateid int, qtysold int, pricepaid decimal(8,2), comment
 VARCHAR(255))
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
WITH SERDEPROPERTIES ('avro.schema.literal'='{\"namespace\": \"dory.sample\",\"name\":
 \"dory_avro\",\"type\": \"record\", \"fields\": [{\"name\":\"salesid\", \"type\":\"int
\"},
{\"name\":\"listid\", \"type\":\"int\"},
{\"name\":\"sellerid\", \"type\":\"int\"},
{\"name\":\"buyerid\", \"type\":\"int\"},
{\"name\":\"eventid\",\"type\":\"int\"},
{\"name\":\"dateid\",\"type\":\"int\"},
{\"name\":\"qtysold\",\"type\":\"int\"},
{\"name\":\"pricepaid\", \"type\": {\"type\": \"bytes\", \"logicalType\": \"decimal\",
 \"precision\": 8, \"scale\": 2}}, {\"name\":\"comment\",\"type\":\"string\"}]}')
STORED AS AVRO
location 's3://mybucket/avro/sales' ;
```

Di seguito viene illustrato un esempio di specificazione dei parametri ROW FORMAT SERDE utilizzando. RegEx

```
create external table spectrum.types(
cbigint bigint,
cbigint_null bigint,
cint int,
cint_null int)
row format serde 'org.apache.hadoop.hive.serde2.RegexSerDe'
with serdeproperties ('input.regex'='([^\\x01]+)\\x01([^\\x01]+)\\x01([^\\x01]+)\
\x01([^\\x01]+)')
stored as textfile
location 's3://mybucket/regex/types';
```

Di seguito viene mostrato un esempio di specifica dei parametri ROW FORMAT SERDE usando Grok.

```
create external table spectrum.grok_log(
timestamp varchar(255),
pid varchar(255),
loglevel varchar(255),
progname varchar(255),
message varchar(255))
row format serde 'com.amazonaws.glue.serde.GrokSerDe'
with serdeproperties ('input.format'='[DFEWI], \\[%{TIMESTAMP_ISO8601:timestamp} #
%{POSINT:pid:int}\\] *(?<loglevel>:DEBUG|FATAL|ERROR|WARN|INFO) -- +%{DATA:progname}:
 %{GREEDYDATA:message}')
stored as textfile
location 's3://mybucket/grok/logs';
```

Il seguente esempio mostra come definire un log di accesso al server Amazon S3 in un bucket S3. È possibile utilizzare Redshift Spectrum per eseguire query sui log degli accessi di Amazon S3.

```
CREATE EXTERNAL TABLE spectrum.mybucket_s3_logs(
bucketowner varchar(255),
bucket varchar(255),
requestdatetime varchar(2000),
remoteip varchar(255),
requester varchar(255),
requested varchar(255),
operation varchar(255),
key varchar(255),
```

```
requesturi_operation varchar(255),
requesturi_key varchar(255),
requesturi_httpprotoversion varchar(255),
httpstatus varchar(255),
errorcode varchar(255),
bytessent bigint,
objectsize bigint,
totaltime varchar(255),
turnaroundtime varchar(255),
referrer varchar(255),
useragent varchar(255),
versionid varchar(255)
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
'input.regex' = '([^ ]*) ([^ ]*) \\[(.*?)\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 \"([^ ]*)\\s*([^ ]*)\\s*([^ ]*)\" (- |[^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\") ([^ ]*).*$')
LOCATION 's3://mybucket/s3logs’;
```

Di seguito viene mostrato un esempio di specifica dei parametri ROW FORMAT SERDE per i file di dati memorizzati nel formato ION.

```
CREATE EXTERNAL TABLE tbl_name (columns)
ROW FORMAT SERDE 'com.amazon.ionhiveserde.IonHiveSerDe'
STORED AS
INPUTFORMAT 'com.amazon.ionhiveserde.formats.IonInputFormat'
OUTPUTFORMAT 'com.amazon.ionhiveserde.formats.IonOutputFormat'
LOCATION 's3://s3-bucket/prefix'
```

Esempi di gestione dei dati

Gli esempi seguenti accedono al file: spi_global_rankings.csv. È possibile caricare il file `spi_global_rankings.csv` in un bucket Amazon S3 per provare questi esempi.

Nell'esempio seguente viene creato lo schema esterno `schema_spectrum_uddh` e il database `spectrum_db_uddh`. Per`aws-account-id`, inserisci l'ID AWS del tuo account e `role-name` inserisci il nome del tuo ruolo Redshift Spectrum.

```
create external schema schema_spectrum_uddh
from data catalog
```

```
database 'spectrum_db_uddh'
iam_role 'arn:aws:iam::aws-account-id:role/role-name'
create external database if not exists;
```

Nell'esempio seguente viene creata la tabella esterna denominata `soccer_league` nello schema esterno `schema_spectrum_uddh`.

```
CREATE EXTERNAL TABLE schema_spectrum_uddh.soccer_league
(
  league_rank smallint,
  prev_rank   smallint,
  club_name   varchar(15),
  league_name varchar(20),
  league_off  decimal(6,2),
  league_def  decimal(6,2),
  league_spi  decimal(6,2),
  league_nspi integer
)
ROW FORMAT DELIMITED
    FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n\l'
stored as textfile
LOCATION 's3://spectrum-uddh/league/'
table properties ('skip.header.line.count'='1');
```

Controllare Il numero di righe nella tabella `soccer_league`.

```
select count(*) from schema_spectrum_uddh.soccer_league;
```

Viene visualizzato il numero di righe.

```
count
645
```

La seguente query visualizza i primi 10 club. Poiché il club `Barcelona` include un carattere non valido nella stringa, per il nome viene visualizzato un valore NULL.

```
select league_rank,club_name,league_name,league_nspi
from schema_spectrum_uddh.soccer_league
where league_rank between 1 and 10;
```

```
league_rank club_name league_name   league_nspi
1  Manchester City Barclays Premier Lea  34595
2  Bayern Munich German Bundesliga  34151
3  Liverpool Barclays Premier Lea  33223
4  Chelsea  Barclays Premier Lea  32808
5  Ajax  Dutch Eredivisie  32790
6  Atletico  Madrid Spanish Primera Divi 31517
7  Real Madrid Spanish Primera Divi  31469
8  NULL         Spanish Primera Divi            31321
9  RB Leipzig German Bundesliga  31014
10  Paris Saint-Ger French Ligue 1   30929
```

Nell'esempio seguente viene modificata la tabella `soccer_league` per specificare le proprietà `invalid_char_handling`, `replacement_char` e `data_cleansing_enabled` della tabella esterna per inserire un punto interrogativo (?) come sostituto di caratteri imprevisti.

```
alter  table schema_spectrum_uddh.soccer_league
set table properties
 ('invalid_char_handling'='REPLACE','replacement_char'='?','data_cleansing_enabled'='true');
```

Nell'esempio seguente viene sottoposta a query la tabella `soccer_league` per team con un grado da 1 a 10.

```
select league_rank,club_name,league_name,league_nspi
from schema_spectrum_uddh.soccer_league
where league_rank between 1 and 10;
```

Poiché le proprietà della tabella sono state modificate, i risultati riportano i primi 10 club, con il carattere di sostituzione punto interrogativo (?) nell'ottava riga per il club `Barcelona`.

```
league_rank club_name league_name  league_nspi
1  Manchester City Barclays Premier Lea 34595
2  Bayern Munich German Bundesliga 34151
3  Liverpool Barclays Premier Lea 33223
4  Chelsea  Barclays Premier Lea 32808
5  Ajax  Dutch Eredivisie 32790
6  Atletico Madrid Spanish Primera Divi 31517
7  Real Madrid Spanish Primera Divi 31469
8  Barcel?na Spanish Primera Divi 31321
9  RB Leipzig German Bundesliga 31014
```

```
10  Paris Saint-Ger French Ligue 1  30929
```

L'esempio seguente modifica la tabella `soccer_league` per specificare le proprietà `invalid_char_handling` della tabella esterna per eliminare le righe con caratteri imprevisti.

```
alter table schema_spectrum_uddh.soccer_league
set table properties
 ('invalid_char_handling'='DROP_ROW','data_cleansing_enabled'='true');
```

Nell'esempio seguente viene sottoposta a query la tabella `soccer_league` per team con un grado da 1 a 10.

```
select league_rank,club_name,league_name,league_nspi
from schema_spectrum_uddh.soccer_league
where league_rank between 1 and 10;
```

I risultati riportano i club principali senza includere l'ottava riga per il `Barcelona`.

```
league_rank   club_name         league_name            league_nspi
1             Manchester City   Barclays Premier Lea   34595
2             Bayern Munich     German Bundesliga      34151
3             Liverpool         Barclays Premier Lea   33223
4             Chelsea           Barclays Premier Lea   32808
5             Ajax              Dutch Eredivisie       32790
6             Atletico Madrid   Spanish Primera Divi   31517
7             Real Madrid       Spanish Primera Divi   31469
9             RB Leipzig        German Bundesliga      31014
10            Paris Saint-Ger   French Ligue 1         30929
```

CREATE EXTERNAL VIEW (anteprima)

Questa è una documentazione di pre-rilascio per le viste del Catalogo dati per Amazon Redshift, che è nella versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche . Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms.

È possibile creare un cluster Amazon Redshift di anteprima per testare nuove funzionalità di Amazon Redshift. Non è possibile utilizzare queste funzionalità in produzione o spostare il cluster di

anteprima in un cluster di produzione o in un cluster su un'altra traccia. Per i termini e le condizioni dell'anteprima, consulta la sezione relativa a beta e anteprime nei termini del servizio AWS.

Come creare un cluster di anteprima

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegli Provisioned clusters dashboard (Pannello di controllo dei cluster con provisioning) e seleziona Clusters (Cluster). Regione AWS Sono elencati i cluster del tuo account nella versione corrente. Nelle colonne dell'elenco è visualizzato un sottoinsieme delle proprietà di ciascun cluster.
3. Nella pagina dell'elenco dei cluster viene visualizzato un banner che introduce l'anteprima. Scegli il pulsante Create preview cluster (Crea cluster di anteprima) per aprire la pagina di creazione del cluster.
4. Inserisci le proprietà del cluster. Scegli la traccia di anteprima che contiene le funzionalità che desideri testare. Consigliamo di assegnare al cluster un nome che indichi che si trova in una traccia di anteprima. Scegli le opzioni per il cluster, tra cui quelle contrassegnate come - anteprima, per le funzionalità che desideri testare. Per informazioni generali sulla creazione di cluster, consulta Creazione di un cluster nella Guida alla gestione di Amazon Redshift.
5. Per creare un cluster di anteprima, scegli Crea cluster.

   > **Note**
   >
   > La traccia `preview_2023` è la traccia di anteprima più recente disponibile. Questa traccia supporta la creazione di cluster solo con tipi di nodo RA3. Il tipo di nodo DC2 e qualsiasi tipo di nodo precedente non sono supportati.

6. Quando il cluster di anteprima è disponibile, utilizza il client SQL per caricare dati ed eseguire query su di essi.

La funzionalità di anteprima delle viste del catalogo dati è disponibile solo nelle seguenti regioni.

- Stati Uniti orientali (Ohio) (us-east-2)
- Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Stati Uniti occidentali (California settentrionale) (us-west-1)
- Asia Pacifico (Tokyo) (ap-northeast-1)

- Europa (Irlanda) (eu-west-1)
- Europa (Stoccolma) (eu-north-1)

Puoi anche creare un gruppo di lavoro di anteprima per testare le viste del catalogo dati. Non è possibile utilizzare queste funzionalità in produzione o trasferire il gruppo di lavoro in un altro gruppo di lavoro. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms. Per istruzioni su come creare un gruppo di lavoro di anteprima, consulta Creazione di un gruppo di lavoro di anteprima.

Crea una vista nel catalogo dati. Una vista del catalogo dati è un singolo schema di visualizzazione che funziona con altri motori SQL come Amazon Athena e Amazon EMR. Puoi eseguire query sulla vista utilizzando il motore che preferisci. Per ulteriori informazioni sulle viste del catalogo dati, consulta Creating Data Catalog views (preview).

Sintassi

```
CREATE EXTERNAL VIEW schema_name.view_name [ IF NOT EXISTS ]
{catalog_name.schema_name.view_name | awsdatacatalog.dbname.view_name |
 external_schema_name.view_name}
AS query_definition;
```

Parametri

schema_name.view_name

Lo schema allegato al AWS Glue database, seguito dal nome della vista.

PROTECTED

Specifica che il comando CREATE EXTERNAL VIEW deve essere completato solo se la query all'interno di query_definition può essere completata correttamente.

IF NOT EXISTS

Crea la vista se non esiste già.

catalog_name.schema_name.view_name | awsdatacatalog.dbname.view_name | external_schema_name.view_name

La notazione dello schema da utilizzare per la creazione della vista. È possibile specificare di utilizzare il AWS Glue Data Catalog, un database Glue creato dall'utente o uno schema

esterno creato dall'utente. Per ulteriori informazioni, consulta CREATE DATABASE e CREATE EXTERNAL SCHEMA.

query_definition

La definizione della query SQL che Amazon Redshift esegue per alterare la vista.

Esempi

L'esempio seguente crea una vista del catalogo dati denominata sample_schema.glue_data_catalog_view.

```
CREATE EXTERNAL PROTECTED VIEW sample_schema.glue_data_catalog_view IF NOT EXISTS
AS SELECT * FROM sample_database.remote_table "remote-table-name";
```

CREATE FUNCTION

Crea una nuova funzione definita dall'utente (UDF) scalare utilizzando una clausola SQL SELECT o un programma Python.

Per maggiori informazioni ed esempi, consulta Creazione di funzioni definite dall'utente.

Privilegi richiesti

È necessario disporre dell'autorizzazione in uno dei seguenti modi per eseguire CREATE OR REPLACE FUNCTION:

- Per CREATE FUNCTION:
  - L'utente con privilegi avanzati può utilizzare linguaggi attendibili e non attendibili per creare funzioni.
  - Gli utenti con il privilegio CREATE [OR REPLACE] FUNCTION possono creare funzioni con linguaggi attendibili.
- Per REPLACE FUNCTION:
  - Superuser
  - Utenti con il privilegio CREATE [OR REPLACE] FUNCTION
  - Proprietario della funzione

Sintassi

```
CREATE [ OR REPLACE ] FUNCTION f_function_name
( { [py_arg_name  py_arg_data_type |
sql_arg_data_type } [ , ... ] ] )
RETURNS data_type
{ VOLATILE | STABLE | IMMUTABLE }
AS $$
  { python_program | SELECT_clause }
$$ LANGUAGE { plpythonu | sql }
```

Parametri

OR REPLACE

Specifica che se una funzione con stesso nome e tipi di dati degli argomenti di input o firma è già esistente, la funzione esistente viene sostituita. Puoi sostituire una funzione solo con una nuova funzione che definisce un set identico di tipi di dati. Devi essere un utente con privilegi avanzati per sostituire una funzione.

Se definisci una funzione con lo stesso nome di una funzione esistente ma con una firma diversa, viene creata una nuova funzione. In altre parole, il nome della funzione è sottoposto a overload. Per ulteriori informazioni, consulta Overload dei nomi delle funzioni.

f_function_name

Il nome della funzione. Se si specifica un nome schema (come `myschema.myfunction`), la funzione viene creata utilizzando lo schema specificato. Altrimenti, la funzione viene creata nello schema corrente. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

Consigliamo di assegnare un prefisso `f_` ai nomi di tutte le funzioni definite dall'utente. Amazon Redshift riserva il prefisso `f_` esclusivamente per le funzioni definite dall'utente e aggiungendo il prefisso `f_` si evita che il nome della funzione entri in conflitto con altri nomi di funzioni SQL predefinite di Amazon Redshift correnti o future. Per ulteriori informazioni, consulta Denominazione delle funzioni definite dall'utente.

Puoi definire più di una funzione con lo stesso nome di funzione se i tipi di dati per gli argomenti di input sono diversi. In altre parole, il nome della funzione è sottoposto a overload. Per ulteriori informazioni, consulta Overload dei nomi delle funzioni.

py_arg_name py_arg_data_type | sql_arg_data type

Per una UDF Python, un elenco di nomi di argomento di input e i tipi di dati. Per una UDF SQL, un elenco di tipi di dati senza nomi di argomento. In una UDF Python, fai riferimento agli argomenti usando i nomi di argomento. In una UDF SQL, fai riferimento agli argomenti che utilizzano $ 1, $ 2 e così via, in base all'ordine degli argomenti nell'elenco degli argomenti.

Per una UDF SQL, i tipi di dati di input e di restituzione possono essere qualsiasi tipo di dati Amazon Redshift standard. Per una UDF Python, il tipo di dati di input e restituiti può essere SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE o TIMESTAMP. Inoltre, le funzioni definite dall'utente (FDU) Python supportano il tipo di dati ANYELEMENT. Questo viene automaticamente convertito in un tipo di dati standard basato sul tipo di dati dell'argomento corrispondente fornito in fase di runtime. Se più argomenti utilizzano ANYELEMENT, verranno risolti tutti nello stesso tipo di dati in fase di runtime, in base al primo argomento ANYELEMENT dell'elenco. Per ulteriori informazioni, consultare Tipi di dati delle funzioni definite dall'utente Python e Tipi di dati.

Puoi specificare un massimo di 32 argomenti.

RETURNS data_type

Il tipo di dati del valore restituito dalla funzione. Il tipo di dati RETURNS può essere qualsiasi tipo di dati standard Amazon Redshift. Inoltre, le UDF Python possono utilizzare un tipo di dati ANYELEMENT, che viene automaticamente convertito in un tipo di dati standard basato sull'argomento fornito in fase di runtime. Se specifichi ANYELEMENT per il tipo di dati di restituzione, almeno un argomento deve utilizzare ANYELEMENT. Il tipo di dati di restituzione effettivo corrisponde al tipo di dati fornito per l'argomento ANYELEMENT quando viene chiamata la funzione. Per ulteriori informazioni, consulta Tipi di dati delle funzioni definite dall'utente Python.

VOLATILE | STABLE | IMMUTABLE

Informa l'ottimizzatore di query circa la volatilità della funzione.

Ottieni la migliore ottimizzazione se etichetti la funzione con la categoria di volatilità più rigida valida. Tuttavia, se la categoria è troppo rigida, c'è il rischio che l'ottimizzatore salti erroneamente alcune chiamate, determinando un set di risultati errato. In ordine di rigidità, a partire dalla meno rigida, le categorie di volatilità sono le seguenti:

- VOLATILE
- STABLE
- IMMUTABLE

VOLATILE

Dati gli stessi argomenti, la funzione può restituire risultati diversi su chiamate successive, anche per le righe in una singola istruzione. L'ottimizzatore di query non può fare alcuna ipotesi sul comportamento di una funzione volatile, quindi una query che utilizza una funzione volatile deve rivalutare la funzione per ogni riga di input.

STABLE

Dati gli stessi argomenti, la funzione è garantita per restituire gli stessi risultati per tutte le righe elaborate all'interno di una singola istruzione. La funzione può restituire risultati diversi se richiamati in istruzioni diverse. Questa categoria consente all'ottimizzatore di ottimizzare più chiamate della funzione all'interno di una singola istruzione in un'unica chiamata per l'istruzione.

IMMUTABLE

Dati gli stessi argomenti, la funzione restituisce sempre lo stesso risultato, per sempre. Quando una query chiama una funzione `IMMUTABLE` con argomenti costanti, l'ottimizzatore prevaluta la funzione.

AS $$ statement $$

Un costrutto che racchiude l'istruzione da eseguire. Le parole chiavi letterali `AS $$` e `$$` sono obbligatorie.

Amazon Redshift richiede di racchiudere l'istruzione nella funzione utilizzando un formato chiamato dollar quoting. Qualsiasi elemento all'interno dell'inquadramento viene trasmesso esattamente com'è. Non è necessario impostare il carattere escape per i caratteri speciali, poiché il contenuto della stringa è scritto letteralmente.

Con dollar quoting, utilizzi una coppia di simboli del dollaro ($$) per indicare l'inizio e la fine dell'istruzione da eseguire, come mostrato nell'esempio seguente.

```
$$ my statement $$
```

Facoltativamente, tra i segni del dollaro in ciascuna coppia, puoi specificare una stringa per aiutare a identificare l'istruzione. La stringa che utilizzi deve essere uguale sia all'inizio che alla fine delle coppie dell'inquadramento. Questa stringa effettua la distinzione tra lettere maiuscole e minuscole e segue gli stessi vincoli di un identificatore senza virgolette, tranne per il fatto che non può contenere segni di dollaro. Gli esempi seguenti utilizzano la stringa `test`.

```
$test$ my statement $test$
```

Per ulteriori informazioni sul dollar quoting, consultare la sezione relativa alle costanti di stringa racchiuse tra simboli del dollaro nell'argomento relativo alla struttura lessicale della documentazione di PostgreSQL.

python_program

Programma Python eseguibile valido che restituisce un valore. L'istruzione che passi con la funzione deve essere conforme ai requisiti di indentazione specificati nella guida di stile per il codice Python sul sito Web Python. Per ulteriori informazioni, consulta Supporto del linguaggio Python per funzioni definite dall'utente.

SQL_clause

Clausola SQL SELECT.

La clausola SELECT non può includere nessuno dei seguenti tipi di clausole:

- FROM
- INTO
- WHERE
- GROUP BY
- ORDER BY
- LIMIT

LANGUAGE { plpythonu | sql }

Per Python, specifica `plpythonu`. Per SQL, specifica `sql`. Devi disporre dell'autorizzazione per l'utilizzo nel linguaggio per SQL o plpythonu. Per ulteriori informazioni, consulta Sicurezza e privilegi dell'UDF.

Note per l'utilizzo

Funzioni nidificate

Puoi chiamare un'altra funzione definita dall'utente (UDF) SQL da una UDF SQL. La funzione nidificata deve esistere quando si esegue il comando CREATE FUNCTION. Amazon Redshift non tiene traccia delle dipendenze per le funzioni definite dall'utente, pertanto se si rimuove la funzione

nidificata, Amazon Redshift non restituisce un errore. Tuttavia, l'UDF non riesce se la funzione nidificata non esiste. Ad esempio, la funzione seguente chiama la funzione `f_sql_greater` nella clausola SELECT.

```
create function f_sql_commission (float, float )
  returns float
stable
as $$
  select f_sql_greater ($1, $2)
$$ language sql;
```

Sicurezza e privilegi dell'UDF

Per creare una UDF devi disporre dell'autorizzazione per l'utilizzo nel linguaggio per SQL o plpythonu (Python). Per impostazione predefinita, USAGE ON LANGUAGE SQL è concesso a PUBLIC. Tuttavia, devi concedere esplicitamente USAGE ON LANGUAGE PLPYTHONU a utenti o gruppi specifici.

Per revocare l'utilizzo per SQL, revoca innanzitutto l'utilizzo da PUBLIC. Quindi concedi l'utilizzo su SQL solo a utenti o gruppi specifici autorizzati a creare UDF SQL. L'esempio seguente revoca l'utilizzo su SQL da PUBLIC, quindi concede l'utilizzo al gruppo di utenti `udf_devs`.

```
revoke usage on language sql from PUBLIC;
grant usage on language sql to group udf_devs;
```

Per eseguire una UDF, devi disporre dell'autorizzazione di esecuzione per ciascuna funzione. Per impostazione predefinita, l'autorizzazione di esecuzione per le nuove UDF è concessa a PUBLIC. Per limitare l'utilizzo, revocare l'autorizzazione da PUBLIC per la funzione. Quindi concedi il privilegio a specifici individui o gruppi.

L'esempio seguente revoca l'esecuzione sulla funzione `f_py_greater` da PUBLIC quindi concede l'utilizzo al gruppo di utenti `udf_devs`.

```
revoke execute on function f_py_greater(a float, b float) from PUBLIC;
grant execute on function f_py_greater(a float, b float) to group udf_devs;
```

Per impostazione predefinita gli utenti con privilegi avanzati hanno tutti i privilegi.

Per ulteriori informazioni, consultare GRANT e REVOKE.

Esempi

Esempio di funzione definita dall'utente Python scalare

L'esempio seguente crea una UDF Python che confronta due numeri interi e restituisce il valore più grande.

```
create function f_py_greater (a float, b float)
  returns float
stable
as $$
  if a > b:
    return a
  return b
$$ language plpythonu;
```

L'esempio seguente esegue una query sulla tabella SALES e chiama la nuova funzione `f_py_greater` per restituire il valore di COMMISSION o il 20 percento di PRICEPAID, a seconda di quale valore è più grande.

```
select f_py_greater (commission, pricepaid*0.20) from sales;
```

Esempio di UDF SQL scalare

L'esempio seguente crea una funzione che confronta due numeri e restituisce il valore più grande.

```
create function f_sql_greater (float, float)
  returns float
stable
as $$
  select case when $1 > $2 then $1
    else $2
  end
$$ language sql;
```

La query seguente chiama la nuova funzione `f_sql_greater` per eseguire una query sulla tabella SALES e restituisce il valore di COMMISSION o il 20 percento di PRICEPAID, a seconda di quale valore è più grande.

```
select f_sql_greater (commission, pricepaid*0.20) from sales;
```

CREATE GROUP

Definisce un nuovo gruppo di utenti. Solo un utente con privilegi avanzati può creare un gruppo.

Sintassi

```
CREATE GROUP group_name
[ [ WITH ] [ USER username ] [, ...] ]
```

Parametri

group_name

Nome del nuovo gruppo di utenti. I nomi di gruppo che iniziano con due caratteri di sottolineatura sono riservati all'uso interno di Amazon Redshift. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

WITH

Sintassi opzionale per indicare i parametri aggiuntivi di CREATE GROUP.

UTENTE

Aggiungi uno o più utenti al gruppo.

username

Nome dell'utente da aggiungere al gruppo.

Esempi

Nell'esempio seguente viene creato un gruppo di utenti denominato ADMIN_GROUP con due utenti, ADMIN1 e ADMIN2.

```
create group admin_group with user admin1, admin2;
```

CREATE IDENTITY PROVIDER

Definisce un nuovo provider di identità. Solo un utente con privilegi avanzati può modificare un provider di identità.

Sintassi

```
CREATE IDENTITY PROVIDER identity_provider_name TYPE type_name
NAMESPACE namespace_name
[PARAMETERS parameter_string]
[APPLICATION_ARN arn]
[IAM_ROLE iam_role]
```

Parametri

identity_provider_name

Il nome del nuovo provider di identità. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

type_name

Il provider di identità con cui interfacciarsi. Azure è attualmente l'unico provider di identità supportato.

namespace_name

Lo spazio dei nomi. Si tratta di un identificatore univoco e in formato abbreviato per la directory del provider di identità.

parameter_string

Stringa contenente un oggetto JSON formattato correttamente che contiene i parametri e i valori richiesti per il provider di identità.

guadagnare

Il nome di risorsa Amazon (ARN) per un'applicazione gestita da IAM Identity Center. Questo parametro è applicabile solo quando il tipo di provider di identità è. AWSIDC

iam_role

Il ruolo IAM che fornisce le autorizzazioni per effettuare la connessione a IAM Identity Center. Questo parametro è applicabile solo quando il tipo di provider di identità è. AWSIDC

Esempi

Nell'esempio seguente viene creato un provider di identità denominato oauth_standard, con TYPE azure, per stabilire una comunicazione con Microsoft Azure Active Directory (AD).

```
CREATE IDENTITY PROVIDER oauth_standard TYPE azure
NAMESPACE 'aad'
PARAMETERS '{"issuer":"https://sts.windows.net/2sdfdsf-d475-420d-b5ac-667adad7c702/",
"client_id":"87f4aa26-78b7-410e-bf29-57b39929ef9a",
"client_secret":"BUAH~ewrqewrqwerUUY^%tHe1oNZShoiU7",
"audience":["https://analysis.windows.net/powerbi/connector/AmazonRedshift"]
}'
```

Puoi connettere un'applicazione gestita da IAM Identity Center con un cluster predisposto esistente o un gruppo di lavoro Amazon Redshift Serverless. Questo ti dà la possibilità di gestire l'accesso a un database Redshift tramite IAM Identity Center. A tale scopo, esegui un comando SQL come nell'esempio seguente. Devi essere un amministratore del database.

```
CREATE IDENTITY PROVIDER "redshift-idc-app" TYPE AWSIDC
NAMESPACE 'awsidc'
APPLICATION_ARN 'arn:aws:sso::123456789012:application/ssoins-12345f67fe123d4/apl-
a0b0a12dc123b1a4'
IAM_ROLE 'arn:aws:iam::123456789012:role/MyRedshiftRole';
```

L'ARN dell'applicazione in questo caso identifica l'applicazione gestita a cui connettersi. Puoi trovarlo eseguendo. `SELECT * FROM SVV_IDENTITY_PROVIDERS;`

Per ulteriori informazioni sull'utilizzo di CREATE IDENTITY PROVIDER, inclusi esempi aggiuntivi, consulta la Native identity provider (IdP) federation for Amazon Redshift (Federazione di gestori dell'identità digitale nativi (IdP) per Amazon Redshift). Per ulteriori informazioni sulla configurazione di una connessione a IAM Identity Center da Redshift, consulta Connect Redshift with IAM Identity Center per offrire agli utenti un'esperienza di single sign-on.

CREATE LIBRARY

Installa una libreria Python che può essere incorporata dagli utenti per la creazione di una funzione definita dall'utente (UDF) con il comando CREATE FUNCTION. La dimensione totale delle librerie installate dall'utente non può superare i 100 MB.

CREATE LIBRARY non può essere eseguito all'interno di un blocco di transazioni (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Amazon Redshift supporta Python versione 2.7. Per ulteriori informazioni, consultare www.python.org.

Per ulteriori informazioni, consulta Importazione di moduli di libreria Python personalizzati.

Privilegi richiesti

Di seguito sono elencati i privilegi richiesti per CREATE LIBRARY:

- Superuser
- Utenti con il privilegio CREATE LIBRARY o con il privilegio del linguaggio specificato

Sintassi

```
CREATE [ OR REPLACE ] LIBRARY library_name LANGUAGE plpythonu
FROM
{ 'https://file_url'
| 's3://bucketname/file_name'
authorization
  [ REGION [AS] 'aws_region']
  IAM_ROLE { default | ‘arn:aws:iam::<Account AWS-id>:role/<role-name>’ }
}
```

Parametri

OR REPLACE

Specifica che se una libreria con lo stesso nome è già esistente, la libreria esistente viene sostituita. REPLACE viene sottoposto immediatamente al commit. Se una UDF che dipende dalla libreria viene eseguita contemporaneamente, l'UDF potrebbe non riuscire o restituire risultati imprevisti, anche se è in esecuzione all'interno di una transazione. Devi essere il proprietario o un utente con privilegi avanzati per sostituire una libreria.

library_name

Il nome della libreria da installare. Non è possibile creare una libreria che contiene un modulo con lo stesso nome di un modulo della libreria standard Python o un modulo Python preinstallato in Amazon Redshift. Se una libreria esistente installata dall'utente utilizza lo stesso pacchetto Python della libreria da installare, è necessario eliminare la libreria esistente prima di installare la nuova libreria. Per ulteriori informazioni, consulta Supporto del linguaggio Python per funzioni definite dall'utente.

LANGUAGE plpythonu

Il linguaggio da usare. Python (plpythonu) è l'unico linguaggio supportato. Amazon Redshift supporta Python versione 2.7. Per ulteriori informazioni, consultare www.python.org.

FROM

La posizione del file della libreria. È possibile specificare il nome di un oggetto e di un bucket Amazon S3 oppure è possibile specificare un URL per scaricare il file da un sito Web pubblico. La libreria deve essere compressa sotto forma di file `.zip`. Per ulteriori informazioni, consultare Creazione e installazione di moduli Python nella documentazione Python.

https://file_url

L'URL per scaricare il file da un sito Web pubblico. L'URL può contenere fino a tre reindirizzamenti. Di seguito è riportato un esempio di un URL.

```
'https://www.example.com/pylib.zip'
```

s3://bucket_name/file_name

Il percorso per un singolo oggetto Amazon S3 che contiene il file della libreria. Di seguito è illustrato un esempio del percorso di un oggetto Amazon S3.

```
's3://mybucket/my-pylib.zip'
```

Se si specifica un bucket Amazon S3, è necessario anche fornire le credenziali per un utente AWS che dispone dell'autorizzazione per scaricare il file.

> **Important**
>
> Se il bucket Amazon S3 non si trova nella stessa AWS regione del cluster Amazon Redshift, devi utilizzare l'opzione REGION per specificare la AWS regione in cui si trovano i dati. Il valore per aws_region deve corrispondere a una AWS regione elencata nella tabella nella descrizione del parametro per il comando COPY. REGION

authorization

Una clausola che indica il metodo che il cluster utilizza per l'autenticazione e l'autorizzazione ad accedere al bucket Amazon S3 contenente il file della libreria. Il cluster deve avere l'autorizzazione ad accedere ad Amazon S3 con le azioni LIST e GET.

La sintassi per l'autorizzazione è uguale a quella del comando COPY. Per ulteriori informazioni, consulta Parametri di autorizzazione.

```
IAM_ROLE { default | ‘arn:aws:iam::<Account AWS-id>:role/<role-name>’
```

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando CREATE LIBRARY.

L'Amazon Resource Name (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione. Se specifichi IAM_ROLE, non è possibile utilizzare ACCESS_KEY_ID e SECRET_ACCESS_KEY, SESSION_TOKEN o CREDENTIALS.

Facoltativamente, se il bucket Amazon S3 utilizza la crittografia lato server, fornisci la chiave di crittografia nella stringa credentials-args. Se usi credenziali di sicurezza temporanee, fornisci il token temporaneo nella stringa credentials-args.

Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee.

REGION [AS] aws_region

La AWS regione in cui si trova il bucket Amazon S3. REGION è obbligatorio quando il bucket Amazon S3 non si trova nella stessa AWS regione del cluster Amazon Redshift. Il valore per aws_region deve corrispondere a una AWS regione elencata nella tabella nella descrizione del REGION parametro per il comando COPY.

Per impostazione predefinita, CREATE LIBRARY presuppone che il bucket Amazon S3 si trovi nella AWS stessa regione del cluster Amazon Redshift.

Esempi

I seguenti due esempi installano il modulo Python urlparse che è compresso in un file denominato `urlparse3-1.0.3.zip`.

Il seguente comando installa una libreria di funzioni definite dall'utente denominata `f_urlparse` da un pacchetto che è stato caricato in un bucket Amazon S3 che si trova nella regione Stati Uniti orientali.

```
create library f_urlparse
language plpythonu
from 's3://mybucket/urlparse3-1.0.3.zip'
credentials 'aws_iam_role=arn:aws:iam::<aws-account-id>:role/<role-name>'
```

```
region as 'us-east-1';
```

L'esempio seguente installa una libreria denominata `f_urlparse` da un file di libreria su un sito Web.

```
create library f_urlparse
language plpythonu
from 'https://example.com/packages/urlparse3-1.0.3.zip';
```

CREATE MASKING POLICY

Crea una nuova policy di mascheramento dinamico dei dati per offuscare i dati di un determinato formato. Per ulteriori informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.

Una policy di mascheramento può essere creata da utenti con privilegi avanzati e da utenti o ruoli che dispongono del ruolo sys:secadmin.

Sintassi

```
CREATE MASKING POLICY
   policy_name [IF NOT EXISTS]
   WITH (input_columns)
   USING (masking_expression);
```

Parametri

nome_policy

Nome della policy di mascheramento. La policy di mascheramento non può avere lo stesso nome di un'altra policy di mascheramento già esistente nel database.

input_columns

Una tupla di nomi di colonne nel formato (col1 type, col2 type...).

I nomi delle colonne vengono utilizzati come input per l'espressione di mascheramento. I nomi delle colonne non devono necessariamente corrispondere ai nomi delle colonne mascherate, ma i tipi di dati di input e di output devono corrispondere.

masking_expression

Espressione SQL utilizzata per trasformare le colonne di destinazione. Può essere scritta utilizzando funzioni di manipolazione dei dati, come le funzioni di manipolazione delle stringhe, o in combinazione con funzioni definite dall'utente scritte in SQL, Python o con AWS Lambda. È possibile includere una tupla di espressioni di colonna per mascherare le policy con più output. Se si utilizza una costante come espressione di mascheramento, è necessario convertirla in modo esplicito su un tipo che corrisponda al tipo di input.

È necessario disporre dell'autorizzazione USAGE per tutte le funzioni definite dall'utente utilizzate nell'espressione di mascheramento.

CREATE MATERIALIZED VIEW

Crea una vista materializzata basata su una o più tabelle Amazon Redshift. Le viste materializzate possono basarsi anche su tabelle esterne create utilizzando Spectrum o una query federata. Per informazioni su Spectrum, vedere Esecuzione di query sui dati esterni utilizzando Amazon Redshift Spectrum. Per informazioni sulla query federata, vedere Esecuzione di query su dati con query federate in Amazon Redshift.

Sintassi

```
CREATE MATERIALIZED VIEW mv_name
[ BACKUP { YES | NO } ]
[ table_attributes ]
[ AUTO REFRESH { YES | NO } ]
AS query
```

Parametri

BACKUP

Una clausola che specifica se la vista materializzata è inclusa negli snapshot automatici o manuali del cluster, che sono memorizzati su Amazon S3.

Il valore predefinito per `BACKUP` è `YES`.

È possibile specificare `BACKUP NO` per risparmiare tempo di elaborazione durante la creazione degli snapshot e il relativo ripristino e per ridurre la quantità di spazio di archiviazione richiesto su Amazon S3.

> **Note**
>
> L'impostazione `BACKUP NO` non ha alcun effetto sulla replica automatica dei dati in altri nodi all'interno del cluster, pertanto le tabelle con `BACKUP NO` specificato vengono ripristinate in caso di problemi con un nodo.

table_attributes

Una clausola che specifica come vengono distribuiti i dati nella vista materializzata, inclusi i seguenti:

- Lo stile della distribuzione della vista materializzata, nel formato `DISTSTYLE { EVEN | ALL | KEY }`. Omettendo questa clausola, la modalità di distribuzione è impostata su `EVEN`. Per ulteriori informazioni, consulta Stili di distribuzione.
- La chiave di distribuzione per la vista materializzata, nel formato `DISTKEY ( distkey_identifier )`. Per ulteriori informazioni, consulta Indicazione degli stili di distribuzione.
- La chiave di ordinamento per la vista materializzata, nel formato `SORTKEY ( column_name [, ...] )`. Per ulteriori informazioni, consulta Utilizzo delle chiavi di ordinamento.

AS query

Una istruzione `SELECT` valida che definisce la vista materializzata e il suo contenuto. Il set di risultati della query definisce le colonne e le righe della vista materializzata. Per informazioni sulle limitazioni durante la creazione di viste materializzate , consultare Limitazioni.

Inoltre, specifici costrutti del linguaggio SQL utilizzati nella query determinano se la vista materializzata può essere aggiornata in modo incrementale o completo. Per informazioni sul metodo di aggiornamento, consultare REFRESH MATERIALIZED VIEW. Per informazioni sulle limitazioni per l'aggiornamento incrementale, consultare Limitazioni per l'aggiornamento incrementale.

Se la query contiene un comando SQL che non supporta l'aggiornamento incrementale, Amazon Redshift visualizza un messaggio che indica che la vista materializzata utilizzerà un aggiornamento completo. Il messaggio può essere visualizzato o meno a seconda dell'applicazione client SQL. Controllare la `state` colonna STV_MV_INFO per visualizzare il tipo di aggiornamento utilizzato da una vista materializzata.

AUTO REFRESH

Una clausola che definisce se la vista materializzata deve essere aggiornata automaticamente con le ultime modifiche apportate dalle tabelle di base. Il valore predefinito è NO. Per ulteriori informazioni, consulta Aggiornamento di una vista materializzata.

Note per l'utilizzo

Per creare una vista materializzata, è necessario disporre dei seguenti privilegi:

- Privilegi CREATE per uno schema.
- Privilegio SELECT a livello di tabella o colonna nelle tabelle di base per creare una vista materializzata. Se si dispone di privilegi a livello di colonna su colonne specifiche, è possibile creare una vista materializzata solo su tali colonne.

Aggiornamento incrementale per le viste materializzate in un datashare

Amazon Redshift supporta l'aggiornamento automatico e incrementale per le viste materializzate in un datashare consumer quando le tabelle di base vengono condivise. L'aggiornamento incrementale è un'operazione in cui Amazon Redshift identifica le modifiche nella tabella o nelle tabelle di base avvenute dopo l'aggiornamento precedente e aggiorna solo i record corrispondenti nella vista materializzata. Questa operazione viene eseguita più rapidamente di un aggiornamento completo e migliora le prestazioni del carico di lavoro. Non è necessario modificare la definizione della vista materializzata per sfruttare l'aggiornamento incrementale.

Esistono alcune limitazioni da notare per sfruttare l'aggiornamento incrementale con una vista materializzata:

- La vista materializzata deve fare riferimento a un solo database, locale o remoto.
- L'aggiornamento incrementale è disponibile solo per le nuove viste materializzate. Pertanto, è necessario eliminare le viste materializzate esistenti e ricrearle affinché si verifichi un aggiornamento incrementale.

Per ulteriori informazioni sulla creazione di viste materializzate in un datashare, consulta Lavorare con le viste nella condivisione dei dati di Amazon Redshift, che contiene diversi esempi di query.

Aggiornamenti DDL a viste materializzate o tabelle di base

Quando si utilizzano viste materializzate in Amazon Redshift, seguire queste note di utilizzo per gli aggiornamenti DDL (Data Definition Language) alle viste materializzate o alle tabelle di base.

- È possibile aggiungere colonne a una tabella di base senza conseguenze sulle viste materializzate che fanno riferimento a tale tabella di base.
- Alcune operazioni possono lasciare la vista materializzata in uno stato in cui non può essere assolutamente aggiornata. Esempi sono le operazioni come la ridenominazione o eliminazione di una colonna, la modifica del tipo di una colonna e la modifica del nome di uno schema. Talli viste materializzate possono essere oggetto di query ma non possono essere aggiornate. In tali casi è necessario eliminare e ricreare la vista materializzata
- In generale, non è possibile modificare la definizione di una vista materializzata (la sua istruzione SQL).
- Non è possibile rinominare una vista materializzata.

Limitazioni

Non è possibile definire una vista materializzata che fa riferimento a uno dei seguenti elementi:

- Viste standard o tabelle e viste di sistema.
- Tabelle temporanee.
- Funzioni definite dall'utente
- La clausola ORDER BY, LIMIT od OFFSET.
- Riferimenti con associazione tardiva alle tabelle di base. In altre parole, tutte le tabelle di base e le relative colonne referenziate nella query SQL di definizione della vista materializzata devono esistere ed essere valide.
- Funzioni applicabili al solo nodo principale: CURRENT_SCHEMA, CURRENT_SCHEMAS, HAS_DATABASE_PRIVILEGE, HAS_SCHEMA_PRIVILEGE, HAS_TABLE_PRIVILEGE.
- Non è possibile utilizzare l'opzione AUTO REFRESH YES quando la definizione della vista materializzata include funzioni mutabili o schemi esterni. Non è inoltre possibile utilizzarla quando si definisce una vista materializzata in un'altra vista materializzata.
- Non è necessario eseguire manualmente ANALYZE su viste materializzate. Attualmente ciò avviene solo tramite AUTO ANALYZE. Per ulteriori informazioni, consulta Analisi delle tabelle.

Esempi

Nell'esempio seguente viene creata una vista materializzata da tre tabelle di base che vengono unite e aggregate. Ogni riga rappresenta una categoria con il numero di biglietti venduti. Quando si esegue una query sulla visualizzazione materializzata tickets_mv, si accede direttamente ai dati precalcolati nella vista materializzata tickets_mv.

```
CREATE MATERIALIZED VIEW tickets_mv AS
    select   catgroup,
    sum(qtysold) as sold
    from     category c, event e, sales s
    where    c.catid = e.catid
    and      e.eventid = s.eventid
    group by catgroup;
```

Nell'esempio seguente viene creata una vista materializzata simile all'esempio precedente e viene utilizzata la funzione di aggregazione MAX().

```
CREATE MATERIALIZED VIEW tickets_mv_max AS
    select   catgroup,
    max(qtysold) as sold
    from     category c, event e, sales s
    where    c.catid = e.catid
    and      e.eventid = s.eventid
    group by catgroup;

SELECT name, state FROM STV_MV_INFO;
```

Nell'esempio seguente viene utilizza una clausola UNION ALL per eseguire il join della tabella `public_sales` di Amazon Redshift e della tabella `spectrum.sales` di Redshift Spectrum per creare una vista materiale `mv_sales_vw`. Per informazioni sul comando CREATE EXTERNAL TABLE per Amazon Redshift Spectrum, consultare CREATE EXTERNAL TABLE. La tabella esterna Redshift Spectrum fa riferimento ai dati su Amazon S3.

```
CREATE MATERIALIZED VIEW mv_sales_vw as
select salesid, qtysold, pricepaid, commission, saletime from public.sales
union all
select salesid, qtysold, pricepaid, commission, saletime from spectrum.sales
```

Nell'esempio seguente viene creata una vista materializzata mv_fq basata su una tabella esterna di query federata. Per informazioni sulla query federata, vedere CREATE EXTERNAL SCHEMA.

```
CREATE MATERIALIZED VIEW mv_fq as select firstname, lastname from apg.mv_fq_example;

select firstname, lastname from mv_fq;
 firstname | lastname
-----------+----------
 John      | Day
 Jane      | Doe
(2 rows)
```

Nell'esempio seguente viene illustrata la definizione di una vista materializzata.

```
SELECT pg_catalog.pg_get_viewdef('mv_sales_vw'::regclass::oid, true);

pg_get_viewdef
-------------------------------------------------
create materialized view mv_sales_vw as select a from t;
```

L'esempio seguente mostra come impostare AUTO REFRESH nella definizione della vista materializzata e specificare DISTSTYLE. Innanzitutto, crea una semplice tabella di base.

```
CREATE TABLE baseball_table (ball int, bat int);
```

Quindi crea una vista materializzata.

```
CREATE MATERIALIZED VIEW mv_baseball DISTSTYLE ALL AUTO REFRESH YES AS SELECT ball AS
 baseball FROM baseball_table;
```

Ora puoi eseguire una query sulla vista materializzata mv_baseball. Per verificare se l'opzione AUTO REFRESH è attivata per una vista materializzata, consulta STV_MV_INFO.

L'esempio seguente crea una vista materializzata che fa riferimento a una tabella di origine in un altro database. Si presuppone che il database contenente la tabella di origine, database_A, si trovi nello stesso cluster o gruppo di lavoro della vista materializzata, creata in database_B. (È possibile sostituire i propri database per l'esempio.) Innanzitutto, crea una tabella in database_A denominata cities con una colonna cityname. Imposta il tipo di dati della colonna come VARCHAR. Dopo aver creato la tabella di origine, esegui il seguente comando in database_B per creare una vista

materializzata la cui origine è la tabella cities. Assicurati di specificare il database e lo schema della tabella di origine nella clausola FROM:

```
CREATE MATERIALIZED VIEW cities_mv AS
SELECT  cityname
FROM    database_A.public.cities;
```

Esegui query nella vista materializzata che hai creato. La query recupera i record la cui origine è la tabella cities nel database_A:

```
select * from cities_mv;
```

Quando esegui l'istruzione SELECT, cities_mv restituisce i record. I record vengono aggiornati dalla tabella di origine solo quando viene eseguita un'istruzione REFRESH. Inoltre, tieni presente che non puoi aggiornare i record direttamente nella vista materializzata. Per informazioni sull'aggiornamento dei dati in una vista materializzata, consulta REFRESH MATERIALIZED VIEW.

Per informazioni dettagliate sulla panoramica della vista materializzata e sui comandi SQL utilizzati per aggiornare e rilasciare le visualizzazioni materializzate, vedere i seguenti argomenti:

- Creazione di viste materializzate in Amazon Redshift
- REFRESH MATERIALIZED VIEW
- DROP MATERIALIZED VIEW

CREATE MODEL

Argomenti

Prerequisiti

Prima di utilizzare l'istruzione CREATE MODEL, completare i prerequisiti descritti in Configurazione del cluster per l'utilizzo di Amazon Redshift ML. Di seguito è riportato un riepilogo approfondito dei prerequisiti.

- Crea un cluster Amazon Redshift con la console di AWS gestione o l'interfaccia a riga di AWS comando (CLI AWS).
- Allega la policy AWS Identity and Access Management (IAM) durante la creazione del cluster.
- Per consentire ad Amazon Redshift e SageMaker assumere il ruolo di interagire con altri servizi, aggiungi la policy di fiducia appropriata al ruolo IAM.

Per maggiori dettagli sul ruolo IAM, sulla policy di attendibilità e su altri prerequisiti, consultare Configurazione del cluster per l'utilizzo di Amazon Redshift ML.

Di seguito, è possibile trovare diversi casi d'uso per l'istruzione CREATE MODEL.

- CREATE MODEL semplice
- CREATE MODEL con guida per l'utente
- Modelli CREATE XGBoost con AUTO OFF
- Bring your own model (BYOM) - inferenza locale
- CREA MODELLO con K-MEANS
- CREATE MODEL completo

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per CREATE MODEL:

- Superuser
- Utenti con il privilegio CREATE MODEL
- Ruoli con il privilegio GRANT CREATE MODEL

Controllo dei costi

Amazon Redshift ML utilizza le risorse cluster esistenti per creare modelli di previsione, quindi non sono previsti costi aggiuntivi. Tuttavia, si potrebbero avere costi aggiuntivi se si deve ridimensionare il

cluster o se si desidera addestrare i propri modelli. Amazon Redshift ML utilizza Amazon SageMaker per addestrare i modelli, il che comporta un costo aggiuntivo associato. Esistono modi per controllare i costi aggiuntivi, come la limitazione della quantità massima di tempo che l'addestramento può richiedere o limitando il numero di esempi di addestramento utilizzati per addestrare il modello. Per ulteriori informazioni consulta Costi per l'utilizzo di Amazon Redshift ML.

CREATE MODEL completo

Di seguito viene riportato un riepilogo delle opzioni di base della sintassi di CREATE MODEL.

Sintassi completa di CREATE MODEL

Di seguito è riportata la sintassi completa dell'istruzione CREATE MODEL.

> **Important**
>
> Quando si crea un modello utilizzando l'istruzione CREATE MODEL, seguire l'ordine delle parole chiave nella sintassi seguente.

```
CREATE MODEL model_name
    FROM { table_name | ( select_statement )  | 'job_name' }
    [ TARGET column_name ]
    FUNCTION function_name ( data_type [, ...] )
    [ RETURNS super ]
    IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>' }
    [ AUTO ON / OFF ]
      -- default is AUTO ON
    [ MODEL_TYPE { XGBOOST | MLP | LINEAR_LEARNER | KMEANS | FORECAST } ]
      -- not required for non AUTO OFF case, default is the list of all supported types
      -- required for AUTO OFF
    [ PROBLEM_TYPE ( REGRESSION | BINARY_CLASSIFICATION | MULTICLASS_CLASSIFICATION ) ]
      -- not supported when AUTO OFF
    [ OBJECTIVE ( 'MSE' | 'Accuracy' | 'F1' | 'F1_Macro' | 'AUC' |
                  'reg:squarederror' | 'reg:squaredlogerror'| 'reg:logistic'|
                  'reg:pseudohubererror' | 'reg:tweedie' | 'binary:logistic' |
 'binary:hinge',
                  'multi:softmax' | 'RMSE' | 'WAPE' | 'MAPE' | 'MASE' |
 'AverageWeightedQuantileLoss' ) ]
      -- for AUTO ON: first 5 are valid
      -- for AUTO OFF: 6-13 are valid
```

```
      -- for FORECAST: 14-18 are valid
    [ PREPROCESSORS 'string' ]
      -- required for AUTO OFF, when it has to be 'none'
      -- optional for AUTO ON
    [ HYPERPARAMETERS { DEFAULT | DEFAULT EXCEPT ( Key 'value' (,...) ) } ]
      -- support XGBoost hyperparameters, except OBJECTIVE
      -- required and only allowed for AUTO OFF
      -- default NUM_ROUND is 100
      -- NUM_CLASS is required if objective is multi:softmax (only possible for AUTO
OFF)
     [ SETTINGS (
       S3_BUCKET 'bucket',  |
         -- required
      TAGS 'string', |
         -- optional
      KMS_KEY_ID 'kms_string', |
         -- optional
      S3_GARBAGE_COLLECT on / off, |
         -- optional, defualt is on.
      MAX_CELLS integer, |
         -- optional, default is 1,000,000
      MAX_RUNTIME integer (, ...) |
         -- optional, default is 5400 (1.5 hours)
      HORIZON integer, |
         -- required if creating a forecast model
      FREQUENCY integer, |
         -- required if creating a forecast model
      PERCENTILES string
         -- optional if creating a forecast model
    ) ]
```

Parametri

model_name

Il nome del modello. Il nome del modello in uno schema deve essere unico.

FROM { table_name | (select_query) | 'job_name'}

Il nome_tabella o la query che specifica i dati di addestramento. Possono essere una tabella esistente nel sistema o una query SELECT compatibile con Amazon Redshift racchiusa tra parentesi, ovvero (). Il risultato della query deve contenere almeno due colonne.

TARGET nome_colonna

Il nome della colonna che diventa la destinazione della previsione. La colonna deve esistere nella clausola FROM.

FUNCTION nome_funzione (tipo_dati [, ...])

Il nome della funzione da creare e i tipi di dati degli argomenti di input. Puoi fornire il nome dello schema di uno schema nel tuo database anziché il nome di una funzione.

RETURNS SUPER (anteprima)

Il tipo di dati del modello da restituire. Il tipo di dati SUPER restituito è applicabile solo ai modelli BYOM remoto.

IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>' }

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando CREAT MODEL. In alternativa, è possibile specificare un ARN di un ruolo IAM per utilizzare quel ruolo.

[AUTO ON / OFF]

Attiva o disattiva il rilevamento automatico CREATE MODEL della selezione di preprocessore, algoritmo e iperparametri. Specificare on quando si crea un modello Forecast indica di utilizzare un AutoPredictor, in cui Amazon Forecast applica le combinazioni ottimali di algoritmi a ogni serie temporale del set di dati.

MODEL_TYPE { XGBOOST | MLP | LINEAR_LEARNER | KMEANS | FORECAST }

(Facoltativo) Specifica il tipo di modello. Puoi specificare se desideri addestrare un modello di un tipo specifico, come XGBoost, perceptron multistrato (MLP), KMEANS o Linear Learner, che sono tutti algoritmi supportati da Amazon Autopilot. SageMaker Se non si specifica il parametro, durante l'addestramento vengono ricercati tutti i tipi di modello supportati per trovare il modello migliore. Puoi anche creare un modello di previsione in Redshift ML per creare previsioni accurate di serie temporali.

PROBLEM_TYPE (REGRESSION | BINARY_CLASSIFICATION | MULTICLASS_CLASSIFICATION)

(Facoltativo) Specifica il tipo di problema. Se si conosce il tipo di problema, è possibile limitare Amazon Redshift alla ricerca solo del modello migliore di quel tipo di modello specifico. Se non si specifica questo parametro, durante l'addestramento viene rilevato un tipo di problema in base ai dati.

OBIETTIVO ('MSE' | 'Precisione' | 'F1' | 'F1Macro' | 'AUC' | 'reg:squarederror' | 'reg:squaredlogerror' | 'reg:logistic' | 'reg:pseudohubererror' | 'reg:tweedie' | 'binary:logistic' | 'binary:hinge' | 'multi:softmax' | 'RMSE' | 'WAPE' | 'MAPE' | 'MASS' | '') AverageWeighted QuantileLoss

(Facoltativo) Specifica il nome del parametro dell'obiettivo utilizzato per misurare la qualità predittiva di un sistema di machine learning. Questo parametro è ottimizzato durante l'addestramento per fornire la migliore stima per i valori dei parametri del modello dai dati. Se non si specifica esplicitamente un parametro, il comportamento di default consiste nell'utilizzare automaticamente MSE: per la regressione, F1: per la classificazione binaria, Accuracy: per la classificazione multiclass. Per ulteriori informazioni sugli obiettivi, consulta AutoML JobObjective nei parametri delle attività Amazon SageMaker API Reference e Learning nella documentazione di XGBOOST. I valori RMSE, WAPE, MAPE, MASE e sono applicabili AverageWeightedQuantileLoss solo ai modelli Forecast. Per ulteriori informazioni, consulta il funzionamento dell'API Predictor. CreateAuto

PREPROCESSORS 'string'

(Facoltativo) Specifica alcune combinazioni di preprocessori per determinati set di colonne. Il formato è un elenco di columnSets e le trasformazioni appropriate da applicare a ciascun set di colonne. Amazon Redshift applica tutti i trasformatori di uno specifico elenco di trasformatori a tutte le colonne dell'elenco corrispondente. ColumnSet Ad esempio, per applicare OneHotEncoder con Imputer alle colonne t1 e t2, usa il comando di esempio seguente.

```
CREATE MODEL customer_churn
FROM customer_data
TARGET 'Churn'
FUNCTION predict_churn
IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>' }
PROBLEM_TYPE BINARY_CLASSIFICATION
OBJECTIVE 'F1'
PREPROCESSORS '[
...
  {"ColumnSet": [
      "t1",
      "t2"
    ],
    "Transformers": [
      "OneHotEncoder",
      "Imputer"
    ]
  },
```

```
  {"ColumnSet": [
      "t3"
    ],
    "Transformers": [
      "OneHotEncoder"
    ]
  },
  {"ColumnSet": [
      "temp"
    ],
    "Transformers": [
      "Imputer",
      "NumericPassthrough"
    ]
  }
]'
SETTINGS (
  S3_BUCKET 'bucket'
)
```

HYPERPARAMETERS { DEFAULT | DEFAULT EXCEPT (key ‘value' (,..)) }

Specifica se i parametri XGBoost di default vengono utilizzati o sostituiti dai valori specificati dall'utente. I valori devono essere racchiusi tra virgolette singole. Di seguito sono riportati esempi di parametri per XGBoost e le loro impostazioni di default.

Nome del parametro	Valore del parametro	Valore predefinito	Note
num_class	Numero intero	Necessario per la classificazione multicl s.	N/D
num_round	Numero intero	100	N/D

Nome del parametro	Valore del parametro	Valore predefinito	Note
tree_method	Stringa	Auto (Automco)	N/D
max_depth	Numero intero	6	[0 , 10]
min_child_weight	Float	1	MinValue: 0,: 120 MaxValue
subsample	Float	1	MinValue: 0,5, MaxValue: 1
gamma	Float	0	MinValue: 0, MaxValue: 5
alpha (alfa)	Float	0	MinValue: 0, MaxValue: 100
eta	Float	0.3	MinValue: 0,1, MaxValue 0,5
colsample_byleve	Float	1	MinValue: 0,1, MaxValue: 1
colsample_bynode	Float	1	MinValue: 0,1, MaxValue: 1
colsample_bytree	Float	1	MinValue: 0,5, MaxValue: 1
lambda	Float	1	MinValue: 0, MaxValue: 100
max_delta_step	Numero intero	0	[0, 10]

SETTINGS (S3_BUCKET 'bucket', | TAGS 'string', | KMS_KEY_ID 'kms_string' , | S3_GARBAGE_COLLECT on / off, | MAX_CELLS integer , | MAX_RUNTIME (,...) , | HORIZON integer, | FREQUENCY forecast_frequency, | PERCENTILES array of strings)

La clausola S3_BUCKET specifica la posizione Amazon S3 utilizzata per archiviare i risultati intermedi.

(Facoltativo) Il parametro TAGS è un elenco separato da virgole di coppie chiave-valore che puoi utilizzare per etichettare le risorse create in Amazon e SageMaker Amazon Forecast. I tag consentono di organizzare le risorse e allocare i costi. I valori nella coppia sono opzionali, quindi puoi creare tag utilizzando il formato `key=value` o semplicemente creando una chiave. Per ulteriori informazioni sui tag in Amazon Redshift, consulta Panoramica dei tag.

(Facoltativo) KMS_KEY_ID specifica se Amazon Redshift utilizza la crittografia lato server con una chiave AWS KMS per proteggere i dati a riposo. I dati in transito sono protetti con Secure Sockets Layer (SSL).

(Facoltativo) S3_GARBAGE_COLLECT { ON | OFF } specifica se Amazon Redshift esegue la rimozione di oggetti inutili (garbage collection) sui set di dati risultanti utilizzati per addestrare i modelli e i modelli stessi. Se impostato su OFF, i set di dati risultanti utilizzati per addestrare i modelli e i modelli stessi rimangono in Amazon S3 e possono essere utilizzati per altri scopi. Se impostato su ON, Amazon Redshift elimina gli artefatti in Amazon S3 al termine dell'addestramento. Il valore di default è ON.

(Facoltativo) MAX_CELLS specifica il numero di celle nei dati di addestramento. Questo valore è il prodotto del numero di record (nella query di addestramento o nella tabella) moltiplicato per il numero di colonne. Il valore di default è 1.000.000.

(Facoltativo) MAX_RUNTIME specifica la durata massima dell'addestramento. I processi di addestramento spesso vengono completati prima di questo valore, a seconda delle dimensioni del set di dati. Specifica la durata massima dell'addestramento. Il valore di default è 5.400 (90 minuti).

HORIZON specifica il numero massimo di previsioni che il modello di previsione può restituire. Una volta addestrato il modello, non è possibile modificare questo numero intero. Questo parametro è obbligatorio se si addestra un modello di previsione.

FREQUENCY specifica la granularità delle unità di tempo desiderate per le previsioni. Le opzioni disponibili sono `Y | M | W | D | H | 30min | 15min | 10min | 5min | 1min`. Questo parametro è obbligatorio se si addestra un modello di previsione.

(Facoltativo) PERCENTILES è una stringa delimitata da virgole che specifica i tipi di previsione utilizzati per addestrare un predittore. I tipi di previsione possono essere quantili da 0,01 a 0,99, con incrementi di 0,01 o superiori. Puoi anche specificare la previsione media con la media. È possibile specificare un massimo di cinque tipi di previsione.

Note per l'utilizzo

Durante l'utilizzo di CREATE MODEL, considerare quanto segue:

- L'istruzione CREATE MODEL opera in modalità asincrona e restituisce i risultati al momento dell'esportazione dei dati di addestramento in Amazon S3. Le fasi rimanenti della formazione in Amazon SageMaker avvengono in background. Mentre l'addestramento è in corso, la funzione di inferenza corrispondente è visibile ma non può essere eseguita. È possibile eseguire query su STV_ML_MODEL_INFO per visualizzare lo stato dell'addestramento.
- L'addestramento può durare fino a 90 minuti in background, per impostazione predefinita nel modello Auto e può essere esteso. Per annullare l'addestramento, è sufficiente eseguire il comando DROP MODEL.
- Il cluster Amazon Redshift utilizzato per creare il modello e il bucket Amazon S3 utilizzato per lo staging dei dati di addestramento e degli artefatti del modello devono trovarsi nella stessa regione. AWS
- Durante la formazione sul modello, Amazon Redshift SageMaker archivia gli artefatti intermedi nel bucket Amazon S3 da te fornito. Per impostazione predefinita, Amazon Redshift esegue la garbage collection alla fine dell'operazione CREATE MODEL. Amazon Redshift rimuove gli oggetti da Amazon S3. Per conservare questi artefatti in Amazon S3, impostare l'opzione S3_GARBAGE COLLECT OFF.
- È necessario utilizzare almeno 500 righe nei dati di addestramento forniti nella clausola FROM.
- Quando si utilizza l'istruzione CREATE MODEL, è possibile specificare fino a 256 colonne di funzionalità (input) nella clausola FROM { table_name | (select_query) }.
- Per AUTO ON, i tipi di colonna che è possibile utilizzare come set di addestramento sono SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE, BOOLEAN, CHAR, VARCHAR, DATE, TIME, TIMETZ, TIMESTAMP e TIMESTAMPTZ. Per AUTO OFF, i tipi di colonna che è possibile utilizzare come set di addestramento sono SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE e BOOLEAN.

- Non è possibile utilizzare DECIMAL, DATE, TIME, TIMETZ, TIMESTAMP, TIMESTAMPTZ, GEOMETRY, GEOGRAPHY HLLSKETCH, SUPER o VARBYTE come tipo di colonna di destinazione.
- Per migliorare la precisione del modello, procedere in uno dei seguenti modi:
  - Aggiungere il maggior numero possibile di colonne rilevanti nel comando CREATE MODEL quando si specificano i dati di addestramento nella clausola FROM.
  - Utilizzare un valore maggiore per MAX_RUNTIVE e MAX_CELLS. Valori maggiori per questo parametro aumentano il costo di addestramento di un modello.
- L'esecuzione dell'istruzione CREATE MODEL viene restituita non appena i dati di addestramento vengono calcolati ed esportati nel bucket Amazon S3. Dopo questo punto, è possibile controllare lo stato dell'addestramento utilizzando il comando SHOW MODEL. Quando un modello addestrato in background non riesce, è possibile controllare l'errore utilizzando SHOW MODEL. Non è possibile riprovare un modello non riuscito. Utilizzare DROP MODEL per rimuovere un modello non riuscito e crearne uno nuovo. Per ulteriori informazioni su SHOW MODEL, consultare SHOW MODEL.
- Il BYOM locale supporta lo stesso tipo di modelli supportati da Amazon Redshift ML per i casi non BYOM. Amazon Redshift supporta XGBoost semplice (utilizzando XGBoost versione 1.0 o successiva), modelli KMEANS senza preprocessori e modelli XGBoost/MLP/Linear Learner addestrati da Amazon Autopilot. SageMaker Supporta quest'ultimo con preprocessori specificati da Autopilot e supportati anche da Amazon Neo. SageMaker
- Se il tuo cluster Amazon Redshift ha un routing avanzato abilitato per il tuo cloud privato virtuale (VPC), assicurati di creare un endpoint VPC Amazon S3 e un endpoint VPC per il VPC in cui si trova il cluster. SageMaker Ciò fa sì che il traffico possa essere eseguito attraverso il VPC tra i servizi durante CREATE MODEL. Per ulteriori informazioni, consulta SageMaker Clarify Job Amazon VPC, sottoreti e gruppi di sicurezza.

Casi d'uso

I seguenti casi d'uso dimostrano come utilizzare CREATE MODEL in base alle proprie esigenze.

CREATE MODEL semplice

Di seguito viene riportato un riepilogo delle opzioni di base della sintassi di CREATE MODEL.

Sintassi di CREATE MODEL semplice

```
CREATE MODEL model_name
    FROM { table_name | ( select_query ) }
```

```
    TARGET column_name
    FUNCTION prediction_function_name
    IAM_ROLE { default }
    SETTINGS (
      S3_BUCKET 'bucket',
      [ MAX_CELLS integer ]
    )
```

Parametri di CREATE MODEL semplice

model_name

Il nome del modello. Il nome del modello in uno schema deve essere unico.

FROM { nome_tabella | (query_select) }

Il nome_tabella o la query che specifica i dati di addestramento. Possono essere una tabella esistente nel sistema o una query SELECT compatibile con Amazon Redshift racchiusa tra parentesi, ovvero (). Il risultato della query deve contenere almeno due colonne.

TARGET nome_colonna

Il nome della colonna che diventa la destinazione della previsione. La colonna deve esistere nella clausola FROM.

FUNCTION nome_funzione_previsione

Un valore che specifica il nome della funzione di machine learning di Amazon Redshift che deve essere generata da CREATE MODEL e utilizzata per effettuare previsioni utilizzando questo modello. La funzione viene creata nello stesso schema dell'oggetto modello e può essere sovraccaricata.

Il machine learning di Amazon Redshift supporta i modelli, come i modelli Xtreme Gradient Boosted Tree (XGBoost) per la regressione e la classificazione.

IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>' }

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando CREAT MODEL. In alternativa, è possibile specificare l'ARN di un ruolo IAM per utilizzare quel ruolo.

S3_BUCKET 'bucket'

Il nome del bucket Amazon S3 creato in precedenza è stato utilizzato per condividere dati e artefatti di addestramento tra Amazon Redshift e. SageMaker Amazon Redshift crea una

sottocartella in questo bucket prima di scaricare i dati di addestramento. Una volta completato l'addestramento, Amazon Redshift elimina la sottocartella creata e il relativo contenuto.

MAX_CELLS integer

Il numero massimo di celle da esportare dalla clausola FROM. Il valore di default è 1.000.000.

Il numero di celle è il prodotto del numero di righe nei dati di addestramento (prodotti dalla tabella o dalla query della clausola FROM) moltiplicato per il numero di colonne. Se il numero di celle nei dati di addestramento è superiore a quello specificato dal parametro max_cells, CREATE MODEL sottocampiona i dati di addestramento della clausola FROM per ridurre le dimensioni del set di addestramento sotto MAX_CELLS. Consentire set di dati di addestramento più grandi può produrre una maggiore precisione, ma può anche significare che il modello richiede più tempo e costa di più.

Per ulteriori informazioni sui costi di utilizzo di Amazon Redshift, consultare Costi per l'utilizzo di Amazon Redshift ML.

Per ulteriori informazioni sui costi associati a vari numeri di cellulare e i dettagli della versione di prova gratuita, consultare Prezzi di Amazon Redshift.

CREATE MODEL con guida per l'utente

Di seguito, è possibile trovare una descrizione delle opzioni per CREATE MODEL in aggiunta alle opzioni descritte in CREATE MODEL semplice.

Per impostazione predefinita, CREATE MODEL cerca la migliore combinazione di preelaborazione e modello per il set di dati specifico. Potrebbe essere necessario un controllo aggiuntivo o introdurre ulteriori informazioni sul dominio (ad esempio il tipo di problema o l'obiettivo) sul modello. In uno scenario di abbandono del cliente, se il risultato "il cliente non è attivo" è raro, l'obiettivo F1 è spesso preferito all'obiettivo Precisione. Poiché i modelli ad alta precisione potrebbero prevedere "il cliente è attivo" per tutto il tempo, ciò si traduce in un'elevata precisione ma poco valore aziendale. Per informazioni sull'obiettivo F1, consulta JobObjectiveAutoML nell' SageMaker Amazon API Reference.

Quindi il CREATE MODEL segue i suggerimenti sugli aspetti specificati, come l'obiettivo. Allo stesso tempo, CREATE MODEL rileva automaticamente i migliori preprocessori e i migliori iperparametri.

Sintassi di CREATE MODEL con guida per l'utente

CREATE MODEL offre una maggiore flessibilità sugli aspetti che è possibile specificare e sugli aspetti che Amazon Redshift rileva automaticamente.

```
CREATE MODEL model_name
    FROM { table_name | ( select_statement ) }
    TARGET column_name
    FUNCTION function_name
    IAM_ROLE { default }
    [ MODEL_TYPE { XGBOOST | MLP | LINEAR_LEARNER} ]
    [ PROBLEM_TYPE ( REGRESSION | BINARY_CLASSIFICATION | MULTICLASS_CLASSIFICATION ) ]
    [ OBJECTIVE ( 'MSE' | 'Accuracy' | 'F1' | 'F1Macro' | 'AUC') ]
    SETTINGS (
      S3_BUCKET 'bucket', |
      S3_GARBAGE_COLLECT { ON | OFF }, |
      KMS_KEY_ID 'kms_key_id', |
      MAX_CELLS integer, |
      MAX_RUNTIME integer (, ...)
    )
```

Parametri di CREATE MODEL con guida per l'utente

MODEL_TYPE { XGBOOST | MLP | LINEAR_LEARNER }

(Facoltativo) Specifica il tipo di modello. Puoi specificare se desideri addestrare un modello di un tipo specifico, come XGBoost, perceptron multistrato (MLP) o Linear Learner, che sono tutti algoritmi supportati da Amazon Autopilot. SageMaker Se non si specifica il parametro, durante l'addestramento vengono ricercati tutti i tipi di modello supportati per trovare il modello migliore.

PROBLEM_TYPE (REGRESSION | BINARY_CLASSIFICATION | MULTICLASS_CLASSIFICATION)

(Facoltativo) Specifica il tipo di problema. Se si conosce il tipo di problema, è possibile limitare Amazon Redshift alla ricerca solo del modello migliore di quel tipo di modello specifico. Se non si specifica questo parametro, durante l'addestramento viene rilevato un tipo di problema in base ai dati.

OBIETTIVO ('MSE' | 'Precisione' | 'F1' | 'F1Macro' | 'AUC')

(Facoltativo) Specifica il nome del parametro dell'obiettivo utilizzato per misurare la qualità predittiva di un sistema di machine learning. Questo parametro è ottimizzato durante l'addestramento per fornire la migliore stima per i valori dei parametri del modello dai dati. Se non si specifica esplicitamente un parametro, il comportamento di default consiste nell'utilizzare automaticamente MSE: per la regressione, F1: per la classificazione binaria, Accuracy: per la classificazione multiclass. Per ulteriori informazioni sugli obiettivi, consulta AutoML JobObjective nell'Amazon SageMaker API Reference.

MAX_CELLS integer

(Facoltativo) Specifica il numero di celle nei dati di addestramento. Questo valore è il prodotto del numero di record (nella query di addestramento o nella tabella) moltiplicato per il numero di colonne. Il valore di default è 1.000.000.

MAX_RUNTIME integer

(Facoltativo) Specifica la durata massima dell'addestramento. I processi di addestramento spesso vengono completati prima di questo valore, a seconda delle dimensioni del set di dati. Specifica la durata massima dell'addestramento. Il valore di default è 5.400 (90 minuti).

S3_GARBAGE_COLLECT { ON | OFF }

(Facoltativo) Specifica se Amazon Redshift esegue la garbage collection sui set di dati risultanti utilizzati per addestrare i modelli e i modelli stessi. Se impostato su OFF, i set di dati risultanti utilizzati per addestrare i modelli e i modelli stessi rimangono in Amazon S3 e possono essere utilizzati per altri scopi. Se impostato su ON, Amazon Redshift elimina gli artefatti in Amazon S3 al termine dell'addestramento. Il valore di default è ON.

KMS_KEY_ID 'kms_key_id'

(Facoltativo) Specifica se Amazon Redshift utilizza la crittografia lato server con una chiave AWS KMS per proteggere i dati a riposo. I dati in transito sono protetti con Secure Sockets Layer (SSL).

PREPROCESSORS 'string'

(Facoltativo) Specifica alcune combinazioni di preprocessori per determinati set di colonne. Il formato è un elenco di columnSets e le trasformazioni appropriate da applicare a ciascun set di colonne. Amazon Redshift applica tutti i trasformatori di uno specifico elenco di trasformatori a tutte le colonne dell'elenco corrispondente. ColumnSet Ad esempio, per applicare OneHotEncoder con Imputer alle colonne t1 e t2, usa il comando di esempio seguente.

```
CREATE MODEL customer_churn
FROM customer_data
TARGET 'Churn'
FUNCTION predict_churn
IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>' }
PROBLEM_TYPE BINARY_CLASSIFICATION
OBJECTIVE 'F1'
PREPROCESSORS '[
...
{"ColumnSet": [
```

```
    "t1",
    "t2"
  ],
  "Transformers": [
    "OneHotEncoder",
    "Imputer"
  ]
},
{"ColumnSet": [
    "t3"
  ],
  "Transformers": [
    "OneHotEncoder"
  ]
},
{"ColumnSet": [
    "temp"
  ],
  "Transformers": [
    "Imputer",
    "NumericPassthrough"
  ]
}
]'
SETTINGS (
S3_BUCKET 'bucket'
)
```

Amazon Redshift supporta i seguenti trasformatori:

- OneHotEncoder — In genere viene utilizzato per codificare un valore discreto in un vettore binario con un valore diverso da zero. Questo trasformatore è adatto per numerosi modelli di machine learning.
- OrdinalEncoder — Codifica i valori discreti in un unico numero intero. Questo trasformatore è adatto per modelli di machine learning specifici, come MLP e Linear Learner.
- NumericPassthrough — Trasmette l'input così com'è nel modello.
- Imputer: riempie i valori mancanti e non i valori numerici (NaN).
- ImputerWithIndicator — Compila i valori mancanti e i valori NaN. Questo trasformatore crea anche un indicatore di eventuali valori mancanti e compilati.

- Normalizzatore: normalizza i valori, il che può migliorare le prestazioni di molti algoritmi di machine learning.
- DateTimeVectorizer — Crea un incorporamento vettoriale, che rappresenta una colonna di tipo di dati datetime che può essere utilizzata nei modelli di machine learning.
- PCA: proietta i dati in uno spazio dimensionale inferiore per ridurre il numero di funzioni mantenendo al contempo il maggior numero di informazioni possibile.
- StandardScaler — Standardizza le funzionalità rimuovendo la media e scalando alla varianza unitaria.
- MinMax — Trasforma le funzionalità ridimensionando ciascuna funzionalità in base a un determinato intervallo.

Amazon Redshift ML memorizza i trasformatori addestrati e li applica automaticamente come parte della query di previsione. Non è necessario specificarli quando si generano le previsioni dal modello.

Modelli CREATE XGBoost con AUTO OFF

L'AUTO OFF CREATE MODEL ha generalmente obiettivi diversi da quelli di default CREATE MODEL.

In qualità di utente avanzato che conosce già il tipo di modello desiderato e gli iperparametri da utilizzare durante l'addestramento di questi modelli, è possibile utilizzare CREATE MODEL con l'opzione AUTO OFF per disattivare il rilevamento automatico CREATE MODEL di preprocessori e iperparametri. A questo scopo, è necessario specificare esplicitamente il tipo di modello. XGBoost al momento è l'unico tipo di modello supportato quando AUTO è impostato su OFF. È possibile specificare iperparametri. Amazon Redshift utilizza i valori predefiniti per tutti gli iperparametri specificati.

Modelli CREATE XGBoost con sintassi AUTO OFF

```
CREATE MODEL model_name
    FROM { table_name | (select_statement ) }
    TARGET column_name
    FUNCTION function_name
    IAM_ROLE { default }
    AUTO OFF
    MODEL_TYPE XGBOOST
    OBJECTIVE { 'reg:squarederror' | 'reg:squaredlogerror' | 'reg:logistic' |
                'reg:pseudohubererror' | 'reg:tweedie' | 'binary:logistic' |
 'binary:hinge' |
```

```
                'multi:softmax' | 'rank:pairwise' | 'rank:ndcg' }
    HYPERPARAMETERS DEFAULT EXCEPT (
        NUM_ROUND '10',
        ETA '0.2',
        NUM_CLASS '10',
        (, ...)
    )
    PREPROCESSORS 'none'
    SETTINGS (
      S3_BUCKET 'bucket', |
      S3_GARBAGE_COLLECT { ON | OFF }, |
      KMS_KEY_ID 'kms_key_id', |
      MAX_CELLS integer, |
      MAX_RUNTIME integer (, ...)
    )
```

Parametri di CREATE XGBoost con AUTO OFF

AUTO OFF

Disattiva il rilevamento automatico CREATE MODEL della selezione di preprocessore, algoritmo e iperparametri.

MODEL_TYPE XGBOOST

Specifica di utilizzare XGBOOST per addestrare il modello.

OBJECTIVE str

Specifica un obiettivo riconosciuto dall'algoritmo. Amazon Redshift supporta reg:squarederror, reg:squaredlogerror, reg:logistic, reg:pseudohubererror, reg:tweedie, binary:logistic, binary:cerniera, multi:softmax. Per ulteriori informazioni su questi obiettivi, consultare.Informazioni sui oarametri di attività nella documentazione di XGBoost.

HYPERPARAMETERS { DEFAULT | DEFAULT EXCEPT (key ‘value' (,..)) }

Specifica se i parametri XGBoost di default vengono utilizzati o sostituiti dai valori specificati dall'utente. I valori devono essere racchiusi tra virgolette singole. Di seguito sono riportati esempi di parametri per XGBoost e le loro impostazioni di default.

Nome del parametro	Valore del parametro	Valore predefinito	Note
num_class	Numero intero	Necessario per la classificazione multicl.	N/D
num_round	Numero intero	100	N/D
tree_method	Stringa	Auto (Automatico)	N/D
max_depth	Numero intero	6	[0 , 10]
min_child_weight	Float	1	MinValue: 0, MaxValue: 120
subsample	Float	1	MinValue: 0,5, MaxValue: 1
gamma	Float	0	MinValue: 0, MaxValue: 5
alpha (alfa)	Float	0	MinValue: 0, MaxValue: 100
eta	Float	0.3	MinValue: 0,1, MaxValue 0,5
colsample_byleve	Float	1	MinValue: 0,1, MaxValue: 1
colsample_bynode	Float	1	MinValue: 0,1, MaxValue: 1

Nome del parametro	Valore del parametro	Valore predefinito	Note
colsample_bytree	Float	1	MinValue: 0,5, MaxValue: 1
lambda	Float	1	MinValue: 0, MaxValue: 100
max_delta_step	Numero intero	0	[0, 10]

L'esempio seguente prepara i dati per XGBoost.

```
DROP TABLE IF EXISTS abalone_xgb;

    CREATE TABLE abalone_xgb (
    length_val float,
    diameter float,
    height float,
    whole_weight float,
    shucked_weight float,
    viscera_weight float,
    shell_weight float,
    rings int,
    record_number int);

    COPY abalone_xgb
    FROM 's3://redshift-downloads/redshift-ml/abalone_xg/'
    REGION 'us-east-1'
    IAM_ROLE default
    IGNOREHEADER 1 CSV;
```

Nell'esempio seguente viene creato un modello XGBoost con opzioni avanzate specificate, quali MODEL_TYPE, OBJECTIVE e PREPROCESSORS.

```
DROP MODEL abalone_xgboost_multi_predict_age;

    CREATE MODEL abalone_xgboost_multi_predict_age
    FROM ( SELECT length_val,
```

```
                  diameter,
                  height,
                  whole_weight,
                  shucked_weight,
                  viscera_weight,
                  shell_weight,
                  rings
   FROM abalone_xgb WHERE record_number < 2500 )
    TARGET rings FUNCTION ml_fn_abalone_xgboost_multi_predict_age
    IAM_ROLE default
    AUTO OFF
    MODEL_TYPE XGBOOST
    OBJECTIVE 'multi:softmax'
    PREPROCESSORS 'none'
    HYPERPARAMETERS DEFAULT EXCEPT (NUM_ROUND '100', NUM_CLASS '30')
    SETTINGS (S3_BUCKET 'your-bucket');
```

Nell'esempio seguente viene utilizzata una query di inferenza per prevedere l'età del pesce con un numero di record maggiore di 2500. Utilizza la funzione ml_fn_abalone_xgboost_multi_predict_age creata dal comando precedente.

```
select ml_fn_abalone_xgboost_multi_predict_age(length_val,
                                                diameter,
                                                height,
                                                whole_weight,
                                                shucked_weight,
                                                viscera_weight,
                                                shell_weight)+1.5 as age
from abalone_xgb where record_number > 2500;
```

Bring your own model (BYOM) - inferenza locale

Amazon Redshift ML supporta l'utilizzo dell'approccio bring your own model (BYOM) nell'inferenza locale.

Di seguito viene riportato un riepilogo delle opzioni per la sintassi di CREATE MODEL per BYOM. Puoi utilizzare un modello addestrato all'esterno di Amazon Redshift con Amazon SageMaker per l'inferenza nel database a livello locale in Amazon Redshift.

Sintassi di CREATE MODEL per l'inferenza locale

Di seguito viene descritta la sintassi CREATE MODEL per l'inferenza locale.

```
CREATE MODEL model_name
    FROM ('job_name' | 's3_path' )
    FUNCTION function_name ( data_type [, ...] )
    RETURNS data_type
    IAM_ROLE { default }
    [ SETTINGS (
      S3_BUCKET 'bucket', | --required
      KMS_KEY_ID 'kms_string') --optional
    ];
```

Amazon Redshift attualmente supporta solo i modelli XGBoost e MLP e Linear Learner preaddestrati per BYOM. Puoi importare SageMaker Autopilot e modelli addestrati direttamente in Amazon SageMaker per l'inferenza locale utilizzando questo percorso.

Parametri di CREATE MODEL per l'inferenza locale

model_name

Il nome del modello. Il nome del modello in uno schema deve essere unico.

FROM ('nome_processo' | 'percorso_s3')

Il job_name utilizza un nome di SageMaker lavoro Amazon come input. Il nome del lavoro può essere un nome di lavoro di SageMaker formazione Amazon o un nome di lavoro Amazon SageMaker Autopilot. Il job deve essere creato nello stesso AWS account proprietario del cluster Amazon Redshift. Per trovare il nome del lavoro, avvia Amazon SageMaker. Nel menu a discesa Training (Addestramento), scegliere Training jobs (Processi di addestramento).

'percorso_s3' specifica la posizione S3 del file di artefatti del modello .tar.gz da utilizzare durante la creazione del modello.

FUNCTION nome_funzione (tipo_dati [, ...])

Il nome della funzione da creare e i tipi di dati degli argomenti di input. È possibile fornire un nome dello schema.

RETURNS data_type

Il tipo di dati del valore restituito dalla funzione.

IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>'}

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando CREATE MODEL.

L'Amazon Resource Name (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione.

SETTINGS (S3_BUCKET 'bucket', | KMS_KEY_ID 'stringa_kms')

La clausola S3_BUCKET specifica la posizione Amazon S3 utilizzata per archiviare i risultati intermedi.

(Facoltativo) La clausola KMS_KEY_ID specifica se Amazon Redshift utilizza la crittografia lato server con una chiave per proteggere i dati inattivi. AWS KMS I dati in transito sono protetti con Secure Sockets Layer (SSL).

Per ulteriori informazioni, consulta CREATE MODEL con guida per l'utente.

Esempio di CREATE MODEL per l'inferenza locale

L'esempio seguente crea un modello che è stato precedentemente addestrato in Amazon SageMaker, al di fuori di Amazon Redshift. Poiché il tipo di modello è supportato da Amazon Redshift ML per l'inferenza locale, il seguente CREATE MODEL crea una funzione che può essere utilizzata in locale in Amazon Redshift. Puoi fornire il nome di un lavoro SageMaker di formazione.

```
CREATE MODEL customer_churn
    FROM 'training-job-customer-churn-v4'
    FUNCTION customer_churn_predict (varchar, int, float, float)
    RETURNS int
    IAM_ROLE default
    SETTINGS (S3_BUCKET 'your-bucket');
```

Dopo aver creato il modello, è possibile utilizzare la funzione customer_churn_predict con i tipi di argomenti specificati per effettuare le previsioni.

Bring your own model (BYOM) - inferenza remota

Amazon Redshift ML supporta anche l'utilizzo dell'approccio bring your own model (BYOM) per l'inferenza remota.

Di seguito viene riportato un riepilogo delle opzioni per la sintassi di CREATE MODEL per BYOM.

> ⚠ Questa è la documentazione non definitiva per il tipo di dati SUPER per l'input nei modelli BYOM in Amazon Redshift ML, che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms.

Se si specifica di utilizzare il tipo di dati SUPER come dati di input e il tipo di dati restituito, si indica che si desidera creare un modello di linguaggio di grandi dimensioni (LLM) ospitato in Amazon. SageMaker JumpStart La creazione di LLM è attualmente disponibile solo come funzionalità di anteprima. Questa anteprima è disponibile di seguito. Regioni AWS

- Stati Uniti orientali (Ohio) (us-east-2)
- Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Asia Pacifico (Tokyo) (ap-northeast-1)
- Europa (Irlanda) (eu-west-1)
- Europa (Stoccolma) (eu-north-1)

È possibile creare un cluster Amazon Redshift di anteprima per testare nuove funzionalità di Amazon Redshift. Non è possibile utilizzare queste funzionalità in produzione o spostare il cluster di anteprima in un cluster di produzione o in un cluster su un'altra traccia. Per i termini e le condizioni dell'anteprima, consulta la sezione relativa a beta e anteprime nei termini del servizio AWS.

Come creare un cluster di anteprima

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegli Provisioned clusters dashboard (Pannello di controllo dei cluster con provisioning) e seleziona Clusters (Cluster). Regione AWS Sono elencati i cluster del tuo account nella versione corrente. Nelle colonne dell'elenco è visualizzato un sottoinsieme delle proprietà di ciascun cluster.
3. Nella pagina dell'elenco dei cluster viene visualizzato un banner che introduce l'anteprima. Scegli il pulsante Create preview cluster (Crea cluster di anteprima) per aprire la pagina di creazione del cluster.

4. Inserisci le proprietà del cluster. Scegli la traccia di anteprima che contiene le funzionalità che desideri testare. Consigliamo di assegnare al cluster un nome che indichi che si trova in una traccia di anteprima. Scegli le opzioni per il cluster, tra cui quelle contrassegnate come - anteprima, per le funzionalità che desideri testare. Per informazioni generali sulla creazione di cluster, consulta Creazione di un cluster nella Guida alla gestione di Amazon Redshift.
5. Per creare un cluster di anteprima, scegli Crea cluster.

> **Note**
>
> La traccia `preview_2023` è la traccia di anteprima più recente disponibile. Questa traccia supporta la creazione di cluster solo con tipi di nodo RA3. Il tipo di nodo DC2 e qualsiasi tipo di nodo precedente non sono supportati.

6. Quando il cluster di anteprima è disponibile, utilizza il client SQL per caricare dati ed eseguire query su di essi.

Puoi anche creare un gruppo di lavoro di anteprima per creare un LLM. Non è possibile utilizzare queste funzionalità in produzione o trasferire il gruppo di lavoro in un altro gruppo di lavoro. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms. Per istruzioni su come creare un gruppo di lavoro di anteprima, consulta Creazione di un gruppo di lavoro di anteprima.

Sintassi di CREATE MODEL per l'inferenza remota

Di seguito viene descritta la sintassi di CREATE MODEL per l'inferenza remota.

```
CREATE MODEL model_name
    FUNCTION function_name ( data_type [, ...] )
    RETURNS data_type
    SAGEMAKER 'endpoint_name'['model_name']
    IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>' }
```

Parametri di CREATE MODEL per l'inferenza remota

model_name

Il nome del modello. Il nome del modello in uno schema deve essere unico.

FUNCTION nome_fn ([tipo_dati] [, ...])

Il nome della funzione e i tipi di dati degli argomenti di input. Consulta Tipi di dati per l'elenco di tutti i tipi di dati supportati. `Geography`, `geometry` e `hllsketch` non sono supportati. Se si specifica di utilizzare il tipo di dati SUPER come dati di input e il tipo di dati restituito, si indica che si desidera creare un modello di linguaggio di grandi dimensioni (LLM) ospitato in Amazon. SageMaker JumpStart

In alternativa, puoi specificare di utilizzare solo il tipo di dati SUPER come dati di input senza utilizzarlo anche come tipo di dati restituito. L'utilizzo del tipo di dati SUPER come input è disponibile solo come funzionalità di anteprima.

È inoltre possibile fornire un nome di schema anziché il nome di una funzione.

RETURNS data_type

Il tipo di dati del valore restituito dalla funzione. Consulta Tipi di dati per l'elenco di tutti i tipi di dati supportati. `Geography`, `geometry` e `hllsketch` non sono supportati. Se si specifica di utilizzare il tipo di dati SUPER come dati di input e il tipo di dati restituito, si indica che si desidera creare un modello di linguaggio di grandi dimensioni (LLM) ospitato in Amazon. SageMaker JumpStart

In alternativa, puoi specificare di utilizzare solo il tipo di dati SUPER come tipo di dati restituito senza utilizzarlo anche come dati di input.

SAGEMAKER 'nome_endpoint'[:'nome_modello']

Il nome dell' SageMaker endpoint Amazon. Se il nome dell'endpoint punta a un endpoint multimodello, aggiungere il nome del modello da utilizzare. L'endpoint deve essere ospitato nella stessa AWS regione del cluster Amazon Redshift. Per trovare il tuo endpoint, avvia Amazon SageMaker. Nel menu a discesa Inference (Inferenza), scegliere Endpoints (Endpoint).

IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>'}

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando CREATE MODEL. In alternativa, è possibile specificare l'ARN di un ruolo IAM per utilizzare quel ruolo.

Quando il modello viene distribuito su un SageMaker endpoint, SageMaker crea le informazioni del modello in Amazon Redshift. Esegue quindi l'inferenza attraverso la funzione esterna. È possibile utilizzare il comando SHOW MODEL per visualizzare le informazioni sul modello sul cluster Amazon Redshift.

Note di utilizzo di CREATE MODEL per l'inferenza remota

Prima di utilizzare CREATE MODEL per l'inferenza remota, considerare quanto segue:

- I modelli BYOM possono supportare un solo argomento con il tipo di dati SUPER come dati di input e output restituito.
- Il modello deve accettare input nel formato di valori separati da virgole (CSV) tramite un tipo di contenuto di testo/CSV in. SageMaker Applicabile solo se non si utilizza il tipo di dati SUPER come input.
- L'endpoint deve essere ospitato dallo stesso AWS account proprietario del cluster Amazon Redshift.
- Gli output dei modelli devono essere un singolo valore del tipo specificato durante la creazione della funzione, nel formato di valori separati da virgole (CSV) tramite un tipo di contenuto testo/CSV in. SageMaker I tipi di dati Varchar non devono essere tra virgolette e ogni output deve essere in una nuova riga. Applicabile solo se hai specificato che il modello non deve restituire il tipo di dati SUPER.
- I modelli accettano valori null come stringhe vuote.
- Assicurati che l' SageMaker endpoint Amazon disponga di risorse sufficienti per accogliere le chiamate di inferenza da Amazon Redshift o che l'endpoint SageMaker Amazon possa essere ridimensionato automaticamente.
- Quando il tipo restituito è SUPER, l'output del modello deve essere JSON e il tipo di contenuto `application/jsonlines`.
- Quando entrambi i tipi di input e output sono SUPER, il modello deve accettare e restituire JSON tramite il tipo di contenuto `application/json`.

Esempio di CREATE MODEL per l'inferenza remota

L'esempio seguente crea un modello che utilizza un SageMaker endpoint per fare previsioni. Assicurarsi che l'endpoint sia in esecuzione per effettuare previsioni e specificarne il nome nel comando CREATE MODEL.

```
CREATE MODEL remote_customer_churn
    FUNCTION remote_fn_customer_churn_predict (varchar, int, float, float)
    RETURNS int
    SAGEMAKER 'customer-churn-endpoint'
    IAM_ROLE default;
```

L'esempio seguente crea un modello linguistico di grandi dimensioni (LLM) utilizzando il tipo di dati SUPER come dati di input e output restituito. I LLM sono ospitati in Jumpstart. SageMaker

```
CREATE MODEL sample_super_data_model
FUNCTION sample_super_data_model_predict(super)
RETURNS super
SAGEMAKER 'sample_super_data_model_endpoint'
IAM_ROLE default;
```

CREA MODELLO con K-MEANS

Amazon Redshift supporta l'algoritmo K-Means che raggruppa i dati che non sono etichettati. Questo algoritmo risolve i problemi di clustering in cui si desidera individuare i raggruppamenti nei dati. I dati non classificati vengono raggruppati e ripartiti in base alle somiglianze e alle differenze.

CREA MODELLO con sintassi K-MEANS

```
CREATE MODEL model_name
    FROM { table_name | ( select_statement ) }
    FUNCTION function_name
    IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>' }
    AUTO OFF
    MODEL_TYPE KMEANS
    PREPROCESSORS 'string'
    HYPERPARAMETERS DEFAULT EXCEPT ( K 'val' [, ...] )
    SETTINGS (
      S3_BUCKET 'bucket',
      KMS_KEY_ID 'kms_string', |
        -- optional
      S3_GARBAGE_COLLECT on / off, |
        -- optional
      MAX_CELLS integer, |
        -- optional
      MAX_RUNTIME integer
        -- optional);
```

CREA MODELLO con i parametri K-MEANS

AUTO OFF

Disattiva il rilevamento automatico CREATE MODEL della selezione di preprocessore, algoritmo e iperparametri.

KMEANS MODEL_TYPE

Specifica di utilizzare XGBOOST per addestrare il modello.

PREPROCESSORS 'string'

Specifica alcune combinazioni di preprocessori per determinati set di colonne. Il formato è un elenco di columnSets e le trasformazioni appropriate da applicare a ciascun set di colonne. Amazon Redshift supporta 3 preprocessori K-Means, vale a dire StandardScaler, e. MinMax NumericPassthrough Se non desideri applicare alcuna preelaborazione per K-Means, scegli NumericPassthrough esplicitamente come trasformatore. Per ulteriori informazioni sulla funzionalità di trasformazione delle proprietà, consultare Parametri di CREATE MODEL con guida per l'utente.

L'algoritmo K-Means utilizza la distanza eucliidea per calcolare la somiglianza. La preelaborazione dei dati garantisce che le caratteristiche del modello rimangano sulla stessa scala e producano risultati affidabili.

HYPERPARAMETERS DEFAULT ECCETTO (K 'val' [,...])

Specifica se vengono utilizzati i parametri K-Means. È necessario specificare il parametro K quando usi l'algoritmo K-Means. Per ulteriori informazioni, consulta K-Means Hyperparameters nella Amazon SageMaker Developer Guide

L'esempio seguente prepara i dati per K-Means.

```
CREATE MODEL customers_clusters
FROM customers
FUNCTION customers_cluster
IAM_ROLE default
AUTO OFF
MODEL_TYPE KMEANS
PREPROCESSORS '[
{
  "ColumnSet": [ "*" ],
  "Transformers": [ "NumericPassthrough" ]
}
]'
HYPERPARAMETERS DEFAULT EXCEPT ( K '5' )
SETTINGS (S3_BUCKET 'bucket');

select customer_id, customers_cluster(...) from customers;
customer_id | customers_cluster
```

```
--------------------
12345              1
12346              2
12347              4
12348              0
```

CREATE MODEL with Forecast

I modelli di previsione in Redshift ML utilizzano Amazon Forecast per creare previsioni accurate di serie temporali. In questo modo puoi utilizzare i dati cronologici relativi a un periodo di tempo per fare previsioni su eventi futuri. I casi d'uso più comuni di Amazon Forecast includono l'utilizzo dei dati dei prodotti al dettaglio per decidere i prezzi dell'inventario, i dati sulla quantità di produzione per prevedere la quantità di un articolo da ordinare e i dati sul traffico Web per prevedere il traffico che potrebbe ricevere un server Web.

Quota limits from Amazon Forecast (Limiti di quota di Amazon Forecast) vengono applicati nei modelli di previsione di Amazon Redshift. Ad esempio, il numero massimo di previsioni è 100, ma è regolabile. L'eliminazione di un modello di previsione non elimina automaticamente le risorse associate in Amazon Forecast. Se elimini un cluster Redshift, vengono eliminati anche tutti i modelli associati.

Tieni presente che i modelli di previsione sono attualmente disponibili solo nelle seguenti regioni:

- Stati Uniti orientali (Ohio) (us-east-2)
- Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Stati Uniti occidentali (Oregon) (us-west-2)
- Asia Pacifico (Mumbai) (ap-south-1)
- Asia Pacifico (Seoul) (ap-northeast-2)
- Asia Pacifico (Singapore) (ap-southeast-1)
- Asia Pacifico (Sydney) (ap-southeast-2)
- Asia Pacifico (Tokyo) (ap-northeast-1)
- Europa (Francoforte) (eu-central-1)
- Europa (Irlanda) (eu-west-1)

CREATE MODEL con sintassi Forecast

```
CREATE [ OR REPLACE ] MODEL forecast_model_name
```

```
FROM { table_name | ( select_query ) }
TARGET column_name
IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>'}
AUTO ON
MODEL_TYPE FORECAST
SETTINGS (
  S3_BUCKET 'bucket',
  HORIZON integer,
  FREQUENCY forecast_frequency
  [PERCENTILES '0.1', '0.5', '0.9']
```

CREATE MODEL con parametri Forecast

forecast_model_name

Il nome del modello. Il nome del modello deve essere univoco.

FROM { nome_tabella | (query_select) }

Il nome_tabella o la query che specifica i dati di addestramento. Può essere una tabella esistente nel sistema o una query SELECT compatibile con Amazon Redshift racchiusa tra parentesi. Il risultato della tabella o della query deve contenere almeno tre colonne: (1) una colonna varchar che specifica il nome della serie temporale. Ogni set di dati può avere più serie temporali, (2) una colonna data/ora e (3) la colonna di destinazione da prevedere. Questa colonna di destinazione deve essere di tipo int o float. Se fornisci un set di dati composto da più di tre colonne, Amazon Redshift presuppone che tutte le colonne aggiuntive facciano parte di una serie temporale correlata. Notare che le serie temporali correlate devono essere di tipo int o float. Per ulteriori informazioni sulle serie temporali correlate, consulta Using Related Time Series Datasets (Utilizzo di set di dati relativi alle serie temporali).

TARGET nome_colonna

Il nome della colonna che diventa la destinazione della previsione. La colonna deve esistere nella clausola FROM.

IAM_ROLE { default | 'arn:aws:iam::<account-id>:role/<role-name>' }

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando CREAT MODEL. In alternativa, è possibile specificare un ARN di un ruolo IAM per utilizzare quel ruolo.

AUTO ON

Attiva il rilevamento automatico CREATE MODEL della selezione di algoritmo e iperparametri. Specificare on quando si crea un modello Forecast indica di utilizzare un Forecast AutoPredictor, in cui Amazon Forecast applica le combinazioni ottimali di algoritmi a ogni serie temporale del set di dati.

MODEL_TYPE FORECAST

Specifica di utilizzare FORECAST per addestrare il modello.

S3_BUCKET 'bucket'

Il nome del bucket Amazon Simple Storage Service creato in precedenza e utilizzato per condividere i dati di addestramento e gli artefatti tra Amazon Redshift e Amazon Forecast. Amazon Redshift crea una sottocartella in questo bucket prima di scaricare i dati di addestramento. Una volta completato l'addestramento, Amazon Redshift elimina la sottocartella creata e il relativo contenuto.

HORIZON integer

Il numero massimo di previsioni che il modello di previsione può restituire. Una volta addestrato il modello, non è possibile modificare questo numero intero.

FREQUENCY forecast_frequency

Specifica la granularità desiderata per le previsioni. Le opzioni disponibili sono `Y | M | W | D | H | 30min | 15min | 10min | 5min | 1min`. Obbligatorio se stai addestrando un modello di previsione.

Stringa PERCENTILES

Una stringa delimitata da virgole che specifica i tipi di previsione utilizzati per addestrare un predittore. I tipi di previsione possono essere quantili da 0,01 a 0,99, con incrementi di 0,01 o superiori. Puoi anche specificare la previsione media con la media. È possibile specificare un massimo di cinque tipi di previsione.

L'esempio seguente dimostra come creare un modello di previsione semplice.

```
CREATE MODEL forecast_example
FROM forecast_electricity_
TARGET target
IAM_ROLE 'arn:aws:iam::<account-id>:role/<role-name>'
AUTO ON
```

```
MODEL_TYPE FORECAST
SETTINGS (S3_BUCKET 'redshift-ml-bucket',
          HORIZON 24,
          FREQUENCY 'H',
          PERCENTILES '0.25,0.50,0.75,mean',
          S3_GARBAGE_COLLECT OFF);
```

Dopo aver creato il modello di previsione, puoi creare una nuova tabella con i dati di previsione.

```
CREATE TABLE forecast_model_results as SELECT Forecast(forecast_example)
```

È quindi possibile eseguire query sulla nuova tabella per ottenere previsioni.

```
SELECT * FROM forecast_model_results
```

CREATE PROCEDURE

Crea una nuova procedura archiviata o sostituisce una procedura esistente per il database corrente.

Per maggiori informazioni ed esempi, consulta Creazione di procedure archiviate in Amazon Redshift.

Privilegi richiesti

È necessario disporre dell'autorizzazione in uno dei seguenti modi per eseguire la PROCEDURA CREATE OR REPLACE:

- Per CREATE PROCEDURE:
  - Superuser
  - Utenti con privilegi CREATE e USAGE sullo schema in cui viene creata la stored procedure
- Per REPLACE PROCEDURE:
  - Superuser
  - Proprietario della procedura

Sintassi

```
CREATE [ OR REPLACE ] PROCEDURE sp_procedure_name
  ( [ [ argname ] [ argmode ] argtype [, ...] ] )
[ NONATOMIC ]
AS $$
```

```
  procedure_body
$$ LANGUAGE plpgsql
[ { SECURITY INVOKER | SECURITY DEFINER } ]
[ SET configuration_parameter { TO value | = value } ]
```

Parametri

OR REPLACE

Una clausola che specifica che se una procedura con stesso nome e tipi di dati degli argomenti di input o firma è già esistente, la procedura esistente viene sostituita. Puoi sostituire una procedura solo con una nuova che definisce un set identico di tipi di dati.

Se definisci una funzione con lo stesso nome di una procedura esistente, ma con una firma diversa, verrà creata una nuova procedura. Ossia, il nome della procedura viene sottoposto a overload. Per ulteriori informazioni, consulta Overload dei nomi delle procedure.

sp_procedure_name

Il nome della procedura. Se specifichi un nome di schema (come **`myschema.myprocedure`**), la procedura viene creata utilizzando lo schema specificato. Altrimenti, la procedura viene creata nello schema corrente. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

Consigliamo di assegnare un prefisso `sp_` ai nomi di tutte le procedure archiviate. Amazon Redshift riserva il prefisso `sp_` solo per le procedure archiviate. Utilizzando il prefisso `sp_`, si assicura che il nome della procedura archiviata non sia in conflitto con alcun nome di procedura archiviata o funzione integrate di Amazon Redshift esistente o futuro. Per ulteriori informazioni, consulta Denominazione delle stored procedure.

Puoi definire più di una procedura con lo stesso nome se i tipi di dati per gli argomenti di input o le firme sono diversi. Ossia, in questo caso il nome della procedura viene sottoposto a overload. Per ulteriori informazioni, consulta Overload dei nomi delle procedure

[argname] [argmode] argtype

Elenco di nomi di argomento, modalità di argomento e tipi di dati. È obbligatorio solo il tipo di dati. Il nome e la modalità sono opzionali e la loro posizione può essere scambiata.

La modalità dell'argomento può essere IN, OUT o INOUT. Il valore predefinito è IN.

Puoi utilizzare gli argomenti OUT e INOUT per restituire uno o più valori dalla chiamata di una procedura. Se sono presenti argomenti OUT o INOUT, la chiamata della procedura restituisce una riga di risultati contenente n colonne, dove n è il numero totale di argomenti OUT o INOUT.

Gli argomenti INOUT sono contemporaneamente argomenti di input e output. Gli argomenti di input includono sia argomenti IN che INOUT, mentre gli argomenti di output includono sia argomenti OUT che INOUT.

Gli argomenti OUT non sono specificati come parte dell'istruzione CALL. Specifica gli argomenti INOUT nell'istruzione CALL della procedura archiviata. Gli argomenti INOUT possono essere utili durante la trasmissione e la restituzione dei valori da una chiamata nidificata e anche durante la restituzione di un `refcursor`. Per ulteriori informazioni sui tipi `refcursor`, consultare Cursori.

I tipi di dati dell'argomento possono essere qualsiasi tipo di dati Amazon Redshift standard. Inoltre, un tipo di dati dell'argomento può essere `refcursor`.

Puoi specificare un massimo di 32 argomenti di input e un massimo di 32 argomenti di output.

AS $$ procedure_body $$

Un costrutto che racchiude la procedura da eseguire. Le parole chiavi letterali AS $$ e $$ sono obbligatorie.

Amazon Redshift richiede di racchiudere l'istruzione nella procedura utilizzando un formato chiamato dollar quoting. Qualsiasi elemento all'interno dell'inquadramento viene trasmesso esattamente com'è. Non è necessario impostare il carattere escape per i caratteri speciali, poiché il contenuto della stringa è scritto letteralmente.

Con dollar quoting, utilizzi una coppia di simboli del dollaro ($$) per indicare l'inizio e la fine dell'istruzione da eseguire, come mostrato nell'esempio seguente.

```
$$ my statement $$
```

Facoltativamente, tra i segni del dollaro in ciascuna coppia, puoi specificare una stringa per aiutare a identificare l'istruzione. La stringa che utilizzi deve essere uguale sia all'inizio che alla fine delle coppie dell'inquadramento. Questa stringa effettua la distinzione tra lettere maiuscole e minuscole e segue gli stessi vincoli di un identificatore senza virgolette, tranne per il fatto che non può contenere segni di dollaro. Gli esempi seguenti utilizzano la stringa test.

```
$test$ my statement $test$
```

Questa sintassi è inoltre utile per il dollar quoting nidificato. Per ulteriori informazioni sul dollar quoting, consultare l'argomento relativo alle costanti di stringa racchiuse tra simboli del dollaro nella sezione relativa alla struttura lessicale della documentazione di PostgreSQL.

procedure_body

Set di istruzioni PL/pgSQL valide. Le istruzioni PL/pgSQL aumentano i comandi SQL con costrutti procedurali, includendo le espressioni di looping e condizionali, per controllare il flusso logico. La maggior parte dei comandi SQL può essere utilizzato nel corpo della procedura, includendo il linguaggio DML (Data Modification Language) come COPY, UNLOAD e INSERT, e il linguaggio DDL (Data Definition Language) come CREATE TABLE. Per ulteriori informazioni, consulta Riferimento al linguaggio PL/pgSQL.

LANGUAGE plpgsql

Valore di un linguaggio. Specifica `plpgsql`. Devi disporre dell'autorizzazione per l'utilizzo nel linguaggio per usare `plpgsql`. Per ulteriori informazioni, consulta GRANT.

NONATOMIC

Crea la procedura archiviata in una modalità di transazione NONATOMIC. La modalità NONATOMIC esegue automaticamente il commit delle istruzioni all'interno della procedura. Inoltre, quando si verifica un errore all'interno della procedura NONATOMIC, l'errore non viene generato nuovamente se viene gestito da un blocco di eccezioni. Per ulteriori informazioni, consulta Gestione delle transazioni e RAISE.

Quando si definisce una procedura archiviata `NONATOMIC`, considerare quanto segue:

- Quando si annidano le chiamate di procedura archiviata, tutte le procedure devono essere create nella stessa modalità di transazione.
- Le opzioni `SECURITY DEFINER` e `SET configuration_parameter` non sono supportate quando si crea una procedura in modalità NONATOMIC.
- Qualsiasi cursore che è aperto (esplicitamente o implicitamente) viene chiuso automaticamente quando viene eseguito un commit implicito. Pertanto, è necessario aprire una transazione esplicita prima di iniziare un ciclo di cursori per garantire che non venga eseguito il commit implicito di qualsiasi codice SQL all'interno dell'iterazione del ciclo.

SECURITY INVOKER | SECURITY DEFINER

L'opzione `SECURITY DEFINER` non è supportata quando è specificato `NONATOMIC`.

Modalità di sicurezza con cui la procedura determina i privilegi di accesso alla procedura in fase di runtime. La procedura deve disporre dell'autorizzazione ad accedere agli oggetti di database sottostanti.

Per la modalità SECURITY INVOKER, la procedura utilizza i privilegi dell'utente che la chiama. L'utente deve disporre di autorizzazioni esplicite per gli oggetti di database sottostanti. L'impostazione predefinita è SECURITY INVOKER.

Per la modalità SECURITY DEFINER, la procedura utilizza i privilegi del proprietario. Il proprietario della procedura è definito come l'utente proprietario della procedura in fase di esecuzione, che non è necessariamente l'utente che ha definito inizialmente la procedura. L'utente che chiama la procedura necessita dei privilegi di esecuzione per la procedura, ma non dei privilegi relativi agli oggetti sottostanti.

SET configuration_parameter { TO value | = value }

Queste opzioni non sono supportate quando è specificato `NONATOMIC`.

La clausola SET implica l'impostazione del parametro `configuration_parameter` specificato sul valore indicato quando viene immessa la procedura. Questa clausola quindi ripristina `configuration_parameter` sul suo valore precedente all'uscita della procedura.

Note per l'utilizzo

Se una procedura archiviata è stata creata utilizzando l'opzione SECURITY DEFINER, quando si richiama la funzione CURRENT_USER dall'interno della procedura archiviata, Amazon Redshift restituisce il nome utente del proprietario della procedura archiviata.

Esempi

> **Note**
>
> Se, durante l'esecuzione di questi esempi, hai riscontrato un errore simile a:
>
> ```
> ERROR: 42601: [Amazon](500310) unterminated dollar-quoted string at or near "$$
> ```
>
> Per informazioni, consultare Panoramica delle procedure archiviate in Amazon Redshift.

L'esempio seguente crea una procedura con due parametri di input.

```
CREATE OR REPLACE PROCEDURE test_sp1(f1 int, f2 varchar(20))
AS $$
DECLARE
```

```
  min_val int;
BEGIN
  DROP TABLE IF EXISTS tmp_tbl;
  CREATE TEMP TABLE tmp_tbl(id int);
  INSERT INTO tmp_tbl values (f1),(10001),(10002);
  SELECT INTO min_val MIN(id) FROM tmp_tbl;
  RAISE INFO 'min_val = %, f2 = %', min_val, f2;
END;
$$ LANGUAGE plpgsql;
```

> **Note**
>
> Quando si scrivono le stored procedure, si consiglia di attenersi a una best practice per proteggere i valori sensibili:
> Non eseguire la codifica fissa delle informazioni sensibili nella logica delle procedure archiviate. Ad esempio, non assegnare una password utente in un'istruzione CREATE USER nel corpo di una procedura archiviata. Ciò rappresenta un rischio per la sicurezza, poiché i valori con codifica fissa possono essere registrati come metadati dello schema nelle tabelle del catalogo. È invece consigliabile passare i valori sensibili, ad esempio le password, come argomenti alla procedura archiviata, mediante parametri.
> Per ulteriori informazioni sulle procedure archiviate, consulta CREATE PROCEDURE e Creazione di procedure archiviate in Amazon Redshift. Per ulteriori informazioni sulle tabelle di catalogo, consulta Tabelle di catalogo di sistema.

L'esempio seguente crea una procedura con un parametro IN, un parametro OUT e un parametro INOUT.

```
CREATE OR REPLACE PROCEDURE test_sp2(f1 IN int, f2 INOUT varchar(256), out_var OUT
 varchar(256))
AS $$
DECLARE
  loop_var int;
BEGIN
  IF f1 is null OR f2 is null THEN
    RAISE EXCEPTION 'input cannot be null';
  END IF;
  DROP TABLE if exists my_etl;
  CREATE TEMP TABLE my_etl(a int, b varchar);
    FOR loop_var IN 1..f1 LOOP
```

```
        insert into my_etl values (loop_var, f2);
        f2 := f2 || '+' || f2;
    END LOOP;
  SELECT INTO out_var count(*) from my_etl;
END;
$$ LANGUAGE plpgsql;
```

CREATE RLS POLICY

Crea una nuova policy di sicurezza a livello di riga per fornire un accesso granulare agli oggetti del database.

Una policy può essere creata da superuser e utenti o ruoli che dispongono del ruolo sys:secadmin.

Sintassi

```
CREATE RLS POLICY policy_name
[ WITH (column_name data_type [, ...]) [ [AS] relation_alias ] ]
USING ( using_predicate_exp )
```

Parametri

nome_policy

Il nome della policy .

WITH (nome_colonna tipo_dati [, ...])

Specifica i valori per nome_colonna e tipo_dati riferiti alle colonne delle tabelle a cui è collegata la policy.

Puoi omettere la clausola WITH solo se la policy RLS non fa riferimento ad alcuna colonna di tabelle a cui è collegata la policy.

AS alias_relazione

Specifica un alias facoltativo per la tabella a cui verrà collegata la policy RLS.

USING (using_predicate_exp)

Specifica un filtro applicato alla clausola WHERE della query. Amazon Redshift applica un predicato di policy prima dei predicati utente a livello di query. Ad esempio, **`current_user = 'joe' and price > 10`** limita Joe a visualizzare solo i record con un prezzo superiore a 10 USD.

Note per l'utilizzo

Quando lavori con l'istruzione CREATE RLS POLICY, tieni presente quanto segue:

- Amazon Redshift supporta filtri che possono far parte di una clausola WHERE di una query.
- Tutte le policy collegate a una tabella devono essere state create con lo stesso alias di tabella.
- Non è necessaria l'autorizzazione SELECT per le tabelle di ricerca. Quando crei una policy, Amazon Redshift concede l'autorizzazione SELECT sulla tabella di ricerca per la rispettiva policy. Una tabella di ricerca è un oggetto tabella utilizzato all'interno di una definizione di policy.
- La sicurezza a livello di riga di Amazon Redshift non supporta i seguenti tipi di oggetto all'interno di una definizione di policy: tabelle di catalogo, relazioni tra database, tabelle esterne, viste regolari, viste con associazione tardiva, tabelle con policy RLS attivate e tabelle temporanee.

Esempi

Le seguenti istruzioni SQL creano le tabelle, gli utenti e i ruoli per l'esempio CREATE RLS POLICY.

```
-- Create users and roles reference in the policy statements.
CREATE ROLE analyst;

CREATE ROLE consumer;

CREATE USER bob WITH PASSWORD 'Name_is_bob_1';

CREATE USER alice WITH PASSWORD 'Name_is_alice_1';

CREATE USER joe WITH PASSWORD 'Name_is_joe_1';

GRANT ROLE sys:secadmin TO bob;

GRANT ROLE analyst TO alice;

GRANT ROLE consumer TO joe;

GRANT ALL ON TABLE tickit_category_redshift TO PUBLIC;
```

Nell'esempio seguente viene creata una policy denominata policy denominata policy_concerts.

```
CREATE RLS POLICY policy_concerts
WITH (catgroup VARCHAR(10))
```

```
USING (catgroup = 'Concerts');
```

CREATE ROLE

Crea un nuovo ruolo personalizzato che è una raccolta di autorizzazioni. Per un elenco di ruoli Amazon Redshift definiti dal sistema, consulta the section called “Ruoli definiti dal sistema di Amazon Redshift”. Esegui una query su SVV_ROLES per visualizzare i ruoli attualmente creati nel cluster o nel gruppo di lavoro.

È prevista una quota per il numero di ruoli che è possibile creare. Per ulteriori informazioni, consulta Quote e limiti in Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Autorizzazioni richieste

Di seguito sono riportati i privilegi richiesti per CREATE ROLE.

- Superuser
- Utenti con il privilegio CREATE ROLE

Sintassi

```
CREATE ROLE role_name
[ EXTERNALID external_id ]
```

Parametri

role_name

Il nome del ruolo. Il nome del ruolo deve essere univoco e non può essere uguale a nessun nome utente. Il nome di un ruolo non può essere una parola riservata.

Un utente con privilegi avanzati o normale con il privilegio CREATE ROLE può creare ruoli. Un utente che non è un utente con privilegi avanzati ma che a cui è stato concesso USAGE al ruolo WITH GRANT OPTION e il privilegio ALTER può concedere questo ruolo a chiunque.

EXTERNALID external_id

L'identificativo del ruolo, associato a un provider di identità. Per ulteriori informazioni, consulta Native identity provider (IdP) federation for Amazon Redshift (Federazione di provider di identità nativi (IdP) per Amazon Redshift).

Esempi

L'esempio seguente crea un ruolo `sample_role1`.

```
CREATE ROLE sample_role1;
```

L'esempio seguente crea un ruolo `sample_role1` con un ID esterno associato a un provider di identità.

```
CREATE ROLE sample_role1 EXTERNALID "ABC123";
```

CREATE SCHEMA

Definisce un nuovo schema per il database corrente.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per CREATE SCHEMA:

- Superuser
- Utenti con il privilegio CREATE SCHEMA

Sintassi

```
CREATE SCHEMA [ IF NOT EXISTS ] schema_name [ AUTHORIZATION username ]
           [ QUOTA {quota [MB | GB | TB] | UNLIMITED} ] [ schema_element [ ... ]

CREATE SCHEMA AUTHORIZATION username[ QUOTA {quota [MB | GB | TB] | UNLIMITED} ]
 [ schema_element [ ... ] ]
```

Parametri

IF NOT EXISTS

Clausola che indica che se lo schema specificato esiste già, il comando non deve apportare modifiche e deve restituire un messaggio che lo schema esiste, piuttosto che terminare con un errore.

Questa clausola è utile durante lo scripting, quindi lo script ha esito positivo se CREATE SCHEMA tenta di creare uno schema già esistente.

schema_name

Nome del nuovo schema. Il nome dello schema non può essere `PUBLIC`. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

> **Note**
>
> L'elenco di schemi nel parametro di configurazione search_path determina la precedenza degli oggetti con nome identico quando a cui viene fatto riferimento senza nome di schema.

AUTHORIZATION

Clausola che concede la proprietà a un utente specificato.

username

Nome del proprietario dello schema.

schema_element

Definizione per uno o più oggetti da creare all'interno dello schema.

QUOTA

La quantità massima di spazio su disco che lo schema specificato può utilizzare. Questo spazio è l'utilizzo del disco collettivo. Include tutte le tabelle permanenti, viste materializzate sotto lo schema specificato e copie duplicate di tutte le tabelle con distribuzione ALL su ciascun nodo di calcolo. La quota dello schema non tiene conto delle tabelle temporanee create come parte di uno spazio dei nomi temporaneo o di uno schema.

Per visualizzare le quote dello schema configurate, consultare SVV_SCHEMA_QUOTA_STATE.

Per visualizzare i record in cui le quote dello schema sono state superate, consultare STL_SCHEMA_QUOTA_VIOLATIONS.

Amazon Redshift converte il valore selezionato in megabyte. Il gigabyte è l'unità di misura predefinita quando non si specifica un valore.

Per impostare e modificare una quota dello schema, è necessario essere un utente con privilegi avanzati del database. Un utente che non è un utente con privilegi avanzati ma che dispone dell'autorizzazione CREATE SCHEMA può creare uno schema con una quota definita. Quando si crea uno schema senza definire una quota, lo schema ha una quota illimitata. Quando si imposta la quota al di sotto del valore corrente utilizzato dallo schema, Amazon Redshift non consente ulteriori operazioni di importazione finché non si libera spazio su disco. Un'istruzione DELETE elimina i dati da una tabella e lo spazio su disco viene liberato solo quando viene eseguita l'istruzione VACUUM.

Amazon Redshift controlla ogni transazione alla ricerca di violazioni delle quote prima di eseguire il commit della transazione. Amazon Redshift controlla le dimensioni (lo spazio su disco utilizzato da tutte le tabelle in uno schema) di ogni schema modificato rispetto alla quota impostata. Poiché il controllo di violazione di quota si verifica alla fine di una transazione, il limite di dimensioni può superare temporaneamente la quota all'interno di una transazione prima del commit. Quando una transazione supera la quota, Amazon Redshift interrompe la transazione, vieta le successive operazioni di importazione e ripristina tutte le modifiche fino a quando non si libera spazio su disco. A causa delle operazioni VACUUM e pulizia interna che avvengono in background, è possibile che uno schema non sia pieno nel momento in cui si controlla lo schema dopo una transazione interrotta.

In via eccezionale, Amazon Redshift ignora la violazione della quota e in alcuni casi esegue il commit delle transazioni. Amazon Redshift esegue questa operazione per le transazioni che consistono esclusivamente in una o più delle seguenti istruzioni in cui non è presente un'istruzione di importazione INSERT o COPY nella stessa transazione:

- DELETE
- TRUNCATE
- VACUUM
- DROP TABLE
- ALTER TABLE APPEND solo quando si spostano i dati dallo schema completo a un altro schema non completo

SENZA LIMITI

Amazon Redshift non impone limiti alla crescita della dimensione totale dello schema.

Limiti

Amazon Redshift applica i seguenti limiti per gli schemi.

- Un massimo di 9900 schemi per database.

Esempi

L'esempio seguente crea uno schema denominato US_SALES e assegna la proprietà all'utente DWUSER.

```
create schema us_sales authorization dwuser;
```

L'esempio seguente crea uno schema denominato US_SALES, assegna la proprietà all'utente DWUSER e imposta la quota su 50 GB.

```
create schema us_sales authorization dwuser QUOTA 50 GB;
```

Per visualizzare il nuovo schema, eseguire una query sulla tabella del catalogo PG_NAMESPACE come mostrato di seguito.

```
select nspname as schema, usename as owner
from pg_namespace, pg_user
where pg_namespace.nspowner = pg_user.usesysid
and pg_user.usename ='dwuser';

   schema |  owner
----------+----------
 us_sales | dwuser
(1 row)
```

L'esempio seguente crea lo schema US_SALES oppure non esegue nulla e restituisce un messaggio se lo schema esiste già.

```
create schema if not exists us_sales;
```

CREATE TABLE

Crea una nuova tabella nel database corrente. Definire un elenco di colonne, ognuna delle quali contiene dati di un tipo distinto. Il proprietario della tabella è l'emittente del comando CREATE TABLE.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per CREATE TABLE:

- Superuser
- Utenti con il privilegio CREATE TABLE

Sintassi

```
CREATE [ [LOCAL ] { TEMPORARY | TEMP } ] TABLE
[ IF NOT EXISTS ] table_name
( { column_name data_type [column_attributes] [ column_constraints ]
  | table_constraints
  | LIKE parent_table [ { INCLUDING | EXCLUDING } DEFAULTS ] }
  [, ... ]  )
[ BACKUP { YES | NO } ]
[table_attributes]

where column_attributes are:
  [ DEFAULT default_expr ]
  [ IDENTITY ( seed, step ) ]
  [ GENERATED BY DEFAULT AS IDENTITY ( seed, step ) ]
  [ ENCODE encoding ]
  [ DISTKEY ]
  [ SORTKEY ]
  [ COLLATE CASE_SENSITIVE | COLLATE CASE_INSENSITIVE  ]

and column_constraints are:
  [ { NOT NULL | NULL } ]
  [ { UNIQUE  |  PRIMARY KEY } ]
  [ REFERENCES reftable [ ( refcolumn ) ] ]

and table_constraints  are:
  [ UNIQUE ( column_name [, ... ] ) ]
  [ PRIMARY KEY ( column_name [, ... ] )  ]
  [ FOREIGN KEY (column_name [, ... ] ) REFERENCES reftable [ ( refcolumn ) ]


and table_attributes are:
  [ DISTSTYLE { AUTO | EVEN | KEY | ALL } ]
  [ DISTKEY ( column_name ) ]
  [ [COMPOUND | INTERLEAVED ] SORTKEY ( column_name [,...]) |  [ SORTKEY AUTO ] ]
```

```
  [ ENCODE AUTO ]
```

Parametri

LOCAL

Facoltativo. Sebbene questa parola chiave sia accettata nell'istruzione, non ha alcun effetto in Amazon Redshift.

TEMPORARY | TEMP

Parola chiave che crea una tabella temporanea visibile solo all'interno della sessione corrente. La tabella viene automaticamente eliminata alla fine della sessione in cui è stata creata. La tabella temporanea può avere lo stesso nome di una tabella permanente. La tabella temporanea viene creata in uno schema separato, specifico per la sessione. Non puoi specificare un nome per questo schema. Questo schema temporaneo diventa il primo schema nel percorso di ricerca, quindi la tabella temporanea ha la precedenza sulla tabella permanente a meno che non si qualifichi il nome della tabella con il nome dello schema per accedere alla tabella permanente. Per ulteriori informazioni sugli schemi e sulla precedenza, consultare search_path.

> **Note**
>
> Per impostazione predefinita, gli utenti del database hanno l'autorizzazione di creare tabelle temporanee tramite la loro appartenenza automatica al gruppo PUBLIC. Per negare questo privilegio a un utente, revoca il privilegio TEMP dal gruppo PUBLIC e quindi concedi esplicitamente il privilegio TEMP solo a utenti o gruppi specifici di utenti.

IF NOT EXISTS

Clausola che indica che se la tabella specificata esiste già, il comando non deve apportare modifiche e deve restituire un messaggio che indica che la tabella esiste, piuttosto che terminare con un errore. Tenere presente che la tabella esistente potrebbe non essere come quella che sarebbe stata creata, per il confronto viene utilizzato solo il nome della tabella.

Questa clausola è utile durante lo scripting, quindi lo script ha esito positivo se CREATE TABLE tenta di creare una tabella già esistente.

table_name

Nome della tabella da creare.

> **Important**
>
> Se specifichi un nome di tabella che inizia con "#", la tabella viene creata come tabella temporanea. Di seguito è riportato un esempio:
>
> ```
> create table #newtable (id int);
> ```
>
> È anche possibile fare riferimento alla tabella con il carattere '# '. Per esempio:
>
> ```
> select * from #newtable;
> ```

La lunghezza massima per il nome della tabella è 127 byte; i nomi più lunghi vengono troncati a 127 byte. Puoi utilizzare caratteri multibyte UTF-8 fino a un massimo di quattro byte. Amazon Redshift applica una quota del numero di tabelle per cluster per tipo di nodo, comprese le tabelle temporanee definite dall'utente e le tabelle temporanee create da Amazon Redshift durante l'elaborazione delle query o la manutenzione del sistema. Facoltativamente, puoi qualificare il nome della tabella con il nome dello schema e del database. Nell'esempio seguente, il nome del database è `tickit`, il nome dello schema è `public` e il nome della tabella è `test`.

```
create table tickit.public.test (c1 int);
```

Se il database o lo schema non esiste, la tabella non viene creata e l'istruzione restituisce un errore. Non è possibile creare tabelle o viste nei database di sistema `template0`, `template1`, `padb_harvest` o `sys:internal`.

Se viene specificato un nome dello schema, la nuova tabella viene creata in quello schema (presupponendo che il creatore abbia accesso allo schema). Il nome della tabella deve essere un nome univoco per lo schema. Se non viene specificato alcuno schema, la tabella viene creata utilizzando lo schema del database corrente. Se stai creando una tabella temporanea, non puoi specificare un nome schema perché le tabelle temporanee esistono in uno schema speciale.

Più tabelle temporanee con lo stesso nome possono esistere contemporaneamente nello stesso database se vengono create in sessioni separate perché le tabelle sono assegnate a schemi diversi. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

column_name

Nome di una colonna da creare nella nuova tabella. La lunghezza massima per il nome della colonna è 127 byte; i nomi più lunghi vengono troncati a 127 byte. Puoi utilizzare caratteri multibyte UTF-8 fino a un massimo di quattro byte. Il numero massimo di colonne che puoi definire in una singola tabella è 1.600. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

> **Note**
>
> Se stai creando una "tabella di grandi dimensioni", assicurati che il tuo elenco di colonne non superi i limiti della larghezza delle righe per i risultati intermedi durante i carichi e l'elaborazione delle query. Per ulteriori informazioni, consulta Note per l'utilizzo.

data_type

Tipo dei dati della colonna da creare. Per le colonne CHAR e VARCHAR, puoi utilizzare la parola chiave MAX invece di dichiarare una lunghezza massima. MAX imposta la lunghezza massima a 4.096 byte per CHAR o a 65.535 byte per VARCHAR. La dimensione massima di un oggetto di tipo GEOMETRY è di 1.048.447 byte.

Per informazioni sui tipi di dati supportati da Amazon Redshift, consultare Tipi di dati.

DEFAULT default_expr

Clausola che assegna un valore di dati predefinito per la colonna. Il tipo di dati di default_expr deve corrispondere al tipo di dati della colonna. Il valore DEFAULT deve essere un'espressione senza variabili. Le sottoquery, i riferimenti incrociati ad altre colonne della tabella corrente e le funzioni definite dall'utente non sono consentiti.

L'espressione default_expr è utilizzata in qualsiasi operazione INSERT che non specifica un valore per la colonna. Se non viene specificato alcun valore predefinito, il valore predefinito per la colonna è null.

Se un'operazione COPY con un elenco di colonne definito omette una colonna che ha un valore DEFAULT, il comando COPY inserisce il valore di default_expr.

IDENTITY(seed, step)

Clausola che specifica che la colonna è una colonna IDENTITY. Una colonna IDENTITY contiene valori univoci generati automaticamente. Il tipo di dati per una colonna IDENTITY deve essere INT o BIGINT.

Quando aggiungi righe utilizzando un'istruzione `INSERT` o `INSERT INTO [tablename] VALUES()`, questi valori iniziano con il valore specificato come seed e si incrementano del numero specificato come step.

Quando carichi la tabella utilizzando un'istruzione `INSERT INTO [tablename] SELECT * FROM` o `COPY`, i dati vengono caricati in parallelo e distribuiti alle sezioni del nodo. Per essere certi che i valori di identità siano univoci, Amazon Redshift salta un numero di valori al momento della creazione dei valori di identità. I valori di identità sono univoci, ma l'ordine potrebbe non corrispondere a quello nel file sorgente.

GENERATED BY DEFAULT AS IDENTITY(seed, step)

Clausola che specifica che la colonna è una colonna IDENTITY predefinita e consente di assegnare automaticamente un valore univoco alla colonna. Il tipo di dati per una colonna IDENTITY deve essere INT o BIGINT. Quando aggiungi righe senza valori, questi valori iniziano con il valore specificato come seed e si incrementano del numero specificato come step. Per informazioni su come i valori vengono generati, consulta IDENTITY.

Inoltre, durante INSERT, UPDATE o COPY pè possibile fornire un valore senza EXPLICIT_IDS. Amazon Redshift utilizza tale valore per inserirlo nella colonna di identità invece di utilizzare il valore generato dal sistema. Il valore può essere un duplicato, un valore minore del seed o un valore compreso tra valori di step. Amazon Redshift non controlla l'univocità dei valori nella colonna. A condizione che il valore non influenzi il prossimo valore generato dal sistema.

> **Note**
>
> Se è richiesta l'univocità nella colonna, non aggiungere un valore duplicato. Aggiungere, invece, un valore univoco che è minore del seed o compreso tra valori di step.

Tenere presente quanto segue riguardo le colonne di identità predefinite:

- Le colonne di identità predefinite sono NON NULL. Non è possibile inserire NULL.
- Per inserire un valore generato in una colonna di identità predefinita, utilizzare la parola chiave `DEFAULT`.

```
INSERT INTO tablename (identity-column-name) VALUES (DEFAULT);
```

- I valori di sostituzione di una colonna di identità predefinita non influenzano il successivo valore generato.
- Non puoi aggiungere una colonna di identità predefinita con l'istruzione ALTER ADD COLUMN.
- Puoi aggiungere una colonna di identità predefinita con l'istruzione ALTER TABLE APPEND.

ENCODE encoding

La codifica della compressione per una colonna. ENCODE AUTO è l'impostazione di default per le tabelle. Amazon Redshift gestisce automaticamente la codifica di compressione per tutte le colonne della tabella. Se si specifica la codifica di compressione per qualsiasi colonna della tabella, la tabella non è più impostata su ENCODE AUTO. Amazon Redshift non gestisce più automaticamente la codifica di compressione per tutte le colonne della tabella. È possibile specificare l'opzione ENCODE AUTO per la tabella per consentire ad Amazon Redshift di gestire automaticamente la codifica di compressione per tutte le colonne della tabella.

Amazon Redshift assegna automaticamente una codifica di compressione iniziale alle colonne per le quali la codifica di compressione non si specifica come segue:

- Per impostazione predefinita, tutte le colonne nelle tabelle temporanee vengono assegnate alla compressione RAW.
- Le colonne definite come chiavi di ordinamento vengono assegnate alla compressione RAW.
- Alle colonne definite come tipi di dati BOOLEAN, REAL, DOUBLE PRECISION, GEOMETRY o GEOGRAPHY viene assegnata la compressione di tipo RAW.
- Le colonne definite come SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIME, TIMETZ, TIMESTAMP o TIMESTAMPTZ sono assegnate alla compressione AZ64.
- Le colonne definite come CHAR, VARCHAR o VARBYTE sono assegnate alla compressione LZO.

> **Note**
>
> Se non vuoi che una colonna venga compressa, specifica esplicitamente la codifica RAW.

Sono supportate le seguenti compression encodings (p. 68):

- AZ64
- BYTEDICT
- DELTA
- DELTA32K
- LZO
- MOSTLY8
- MOSTLY16
- MOSTLY32
- RAW (nessuna compressione)
- RUNLENGTH
- TEXT255
- TEXT32K
- ZSTD

DISTKEY

Parola chiave che specifica che la colonna è la chiave di distribuzione per la tabella. Solo una colonna in una tabella può essere la chiave di distribuzione. Puoi utilizzare la parola chiave DISTKEY dopo il nome di una colonna o come parte della definizione della tabella utilizzando la sintassi DISTKEY (column_name). Entrambi i metodi hanno lo stesso effetto. Per ulteriori informazioni, vedi il parametro DISTSTYLE più avanti in questo argomento.

Il tipo di dati di una colonna chiave di distribuzione può essere: BOOLEAN, REAL, DOUBLE PRECISION, SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIME, TIMETZ, TIMESTAMP o TIMESTAMPTZ, CHAR o VARCHAR.

SORTKEY

Parola chiave che specifica che la colonna è la chiave di ordinamento per la tabella. Quando vengono caricati nella tabella, i dati vengono ordinati in base a una o più colonne designate come chiavi di ordinamento. Puoi utilizzare la parola chiave SORTKEY dopo il nome di una colonna per specificare una chiave di ordinamento a colonna singola oppure puoi specificare una o più colonne come colonne chiave di ordinamento per la tabella utilizzando la sintassi SORTKEY (column_name [, ...]). Solo le chiavi di ordinamento composte vengono create con questa sintassi.

Puoi definire un massimo di 400 colonne SORTKEY per tabella.

Il tipo di dati di una colonna della chiave di ordinamento può essere: BOOLEAN, REAL, DOUBLE PRECISION, SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIME, TIMETZ, TIMESTAMP o TIMESTAMPTZ, CHAR o VARCHAR.

COLLATE CASE_SENSITIVE | COLLATE CASE_INSENSITIVE

Una clausola che specifica se la stringa di ricerca o il confronto sulla colonna è CASE_SENSITIVE o CASE_INSENSITIVE. Il valore di default corrisponde alla stessa configurazione corrente della distinzione tra maiuscole e minuscole del database.

Per informazioni sul confronto di database, utilizzare il comando seguente:

```
SELECT db_collation();

db_collation
----------------
 case_sensitive
(1 row)
```

NOT NULL | NULL

NOT NULL specifica che la colonna non può contenere valori null. NULL, il valore predefinito, specifica che la colonna accetta valori nulli. Le colonne IDENTITY sono dichiarate NOT NULL per impostazione predefinita.

UNIQUE

Parola chiave che specifica che la colonna può contenere solo valori univoci. Il comportamento del vincolo di tabella univoco è uguale a quello dei vincoli di colonna, con la possibilità di estendersi su più colonne. Per definire un vincolo di tabella univoco, utilizza la sintassi UNIQUE (column_name [, ...]).

> **Important**
>
> I vincoli univoci sono informativi e non vengono applicati dal sistema.

PRIMARY KEY

Parola chiave che specifica che la colonna è la chiave primaria per la tabella. È possibile definire solo una colonna come chiave primaria utilizzando una definizione di colonna. Per definire un

vincolo di tabella con una chiave primaria a più colonne, utilizza la sintassi PRIMARY KEY (column_name [, ...]).

L'identificazione di una colonna come chiave primaria fornisce i metadati relativi alla progettazione dello schema. Una chiave primaria implica che altre tabelle possono fare affidamento su questo set di colonne come identificatore univoco per le righe. È possibile specificare una chiave primaria per una tabella, sia come vincolo di colonna sia come vincolo di tabella. Il vincolo di chiave primaria deve nominare un set di colonne diverso da altri set di colonne nominati da qualsiasi vincolo univoco definito per la stessa tabella.

Le colonne PRIMARY KEY sono anche definite come NOT NULL.

> **Important**
>
> I vincoli della chiave primaria sono solo informativi. Non vengono applicati dal sistema, ma vengono utilizzati dal pianificatore.

References reftable [(refcolumn)]

Clausola che specifica un vincolo di chiave esterna che implica che la colonna deve contenere solo valori che corrispondono ai valori nella colonna di riferimento di alcune righe della tabella di riferimento. Le colonne di riferimento devono essere le colonne di un vincolo di chiave primaria o univoco nella tabella di riferimento.

> **Important**
>
> I vincoli della chiave esterna sono solo informativi. Non vengono applicati dal sistema, ma vengono utilizzati dal pianificatore.

LIKE parent_table [{ INCLUDING | EXCLUDING } DEFAULTS]

Clausola che specifica una tabella esistente da cui la nuova tabella copia automaticamente i nomi delle colonne, i tipi di dati e i vincoli NOT NULL. La nuova tabella e la tabella padre vengono disassociate e tutte le modifiche apportate alla tabella padre non vengono applicate alla nuova tabella. Le espressioni predefinite per le definizioni delle colonne copiate vengono copiate solo se è specificato INCLUDING DEFAULTS. Il comportamento predefinito consiste nell'escludere le espressioni predefinite, in modo che tutte le colonne della nuova tabella abbiano valori predefiniti null.

Le tabelle create con l'opzione LIKE non ereditano i vincoli di chiave primaria ed esterna. Lo stile di distribuzione, le chiavi di ordinamento e le proprietà BACKUP e NULL vengono ereditate dalle tabelle LIKE, ma non è possibile impostarle in modo esplicito nell'istruzione CREATE TABLE ... Istruzione LIKE.

BACKUP { YES | NO }

Clausola che specifica se la tabella deve essere inclusa negli snapshot di cluster automatizzati e manuali. Per le tabelle come le tabelle di gestione temporanea che non contengono dati critici, specificare BACKUP NO per risparmiare tempo di elaborazione durante la creazione di snapshot e il ripristino da snapshot e per ridurre lo spazio di archiviazione in Amazon Simple Storage Service. L'impostazione BACKUP NO non ha alcun effetto sulla replica automatica dei dati in altri nodi all'interno del cluster, pertanto le tabelle con BACKUP NO specificato vengono ripristinate con un errore del nodo. L'impostazione predefinita è BACKUP YES.

DISTSTYLE { AUTO | EVEN | KEY | ALL }

Parola chiave che definisce lo stile di distribuzione dei dati per l'intera tabella. Amazon Redshift distribuisce le righe di una tabella nei nodi di calcolo a seconda dello stile di distribuzione specificato per la tabella. Il valore predefinito è AUTO.

Lo stile di distribuzione selezionato per le tabelle influisce sulle prestazioni generali del database. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati. Gli stili di distribuzione possibili sono i seguenti:

- AUTO: Amazon Redshift assegna uno stile di distribuzione ottimale basato sui dati della tabella. Ad esempio, se viene specificato lo stile di distribuzione AUTO, Amazon Redshift assegna inizialmente la distribuzione ALL a una tabella di piccole dimensioni. Quando le dimensioni della tabella aumentano, Amazon Redshift potrebbe modificare lo stile di distribuzione in KEY, scegliendo la chiave primaria (o una colonna della chiave primaria composita) come DISTKEY. Se le dimensioni della tabella aumentano e nessuna delle colonne è adatta per essere utilizzata come DISTKEY, Amazon Redshift modifica lo stile di distribuzione in EVEN. La modifica dello stile di distribuzione avviene in background con un impatto minimo sulle query degli utenti.

  Per visualizzare lo stile di distribuzione applicato a una tabella, eseguire una query sulla tabella del catalogo di sistema PG_CLASS. Per ulteriori informazioni, consulta Visualizzazione degli stili di distribuzione.
- EVEN: i dati della tabella sono distribuiti uniformemente tra i nodi di un cluster in una distribuzione round robin. Gli ID riga vengono utilizzati per determinare la distribuzione e approssimativamente lo stesso numero di righe viene distribuito a ciascun nodo.

- KEY: i dati sono distribuiti dai valori nella colonna DISTKEY. Quando imposti le colonne di unione delle tabelle di unione come chiavi di distribuzione, le righe di unione di entrambe le tabelle sono collocate nei nodi di calcolo. Quando i dati sono collocati, l'ottimizzatore può eseguire i join in modo più efficiente. Se specifichi DISTSTYLE KEY, devi nominare una colonna DISTKEY per la tabella o come parte della definizione della colonna. Per ulteriori informazioni, vedi il precedente parametro DISTKEY in questo argomento.
- ALL: copia dell'intera tabella viene distribuita a ogni nodo. Questo stile di distribuzione garantisce che tutte le righe richieste per ogni join siano disponibili su ogni nodo, ma moltiplica i requisiti di storage e aumenta i tempi di caricamento e manutenzione per la tabella. La distribuzione ALL può migliorare i tempi di esecuzione quando viene utilizzata con determinate tabelle di dimensioni in cui la distribuzione KEY non è appropriata, ma i miglioramenti delle prestazioni devono essere confrontati con i costi di manutenzione.

DISTKEY (column_name)

Vincolo che specifica che la colonna da usare come chiave di distribuzione per la tabella. Puoi utilizzare la parola chiave DISTKEY dopo il nome di una colonna o come parte della definizione della tabella utilizzando la sintassi DISTKEY (column_name). Entrambi i metodi hanno lo stesso effetto. Per ulteriori informazioni, vedi il precedente parametro DISTSTYLE in questo argomento.

[COMPOUND | INTERLEAVED] SORTKEY (nome_colonna [,...]) | [SORTKEY AUTO]

Specifica una o più chiavi di ordinamento per la tabella. Quando vengono caricati nella tabella, i dati vengono ordinati in base alle colonne designate come chiavi di ordinamento. Puoi utilizzare la parola chiave SORTKEY dopo il nome di una colonna per specificare una chiave di ordinamento a colonna singola oppure puoi specificare una o più colonne come colonne chiave di ordinamento per la tabella utilizzando la sintassi `SORTKEY (column_name [ , ... ] )`.

È possibile specificare facoltativamente lo stile di ordinamento COMPOUND o INTERLEAVED. Se si specifica SORTKEY con colonne, l'impostazione di default è COMPOSE. Per ulteriori informazioni, consulta Utilizzo delle chiavi di ordinamento.

Se non si specifica alcuna opzione per le chiavi di ordinamento, il valore di default è AUTO.

È possibile definire un massimo di 400 colonne COMPOUND SORTKEY o 8 colonne INTERLEAVED SORTKEY per tabella.

AUTO

Specifica che Amazon Redshift assegna una chiave di ordinamento ottimale basata sui dati della tabella. Ad esempio, se viene specificata la chiave di ordinamento AUTO, Amazon

Redshift inizialmente non assegna alcuna chiave di ordinamento a una tabella. Se Amazon Redshift determina che una chiave di ordinamento migliorerà le prestazioni delle query, Amazon Redshift potrebbe modificare la chiave di ordinamento della tabella. L'ordinamento effettivo della tabella viene eseguito dall'ordinamento automatico della tabella. Per ulteriori informazioni, consulta Ordinamento automatico delle tabelle.

Amazon Redshift non modifica le tabelle con chiavi di ordinamento o di distribuzione esistenti. Con un'eccezione, se una tabella ha una chiave di distribuzione che non è mai stata utilizzata in un JOIN, la chiave potrebbe essere modificata se Amazon Redshift determina che esiste una chiave migliore.

Per visualizzare la chiave di ordinamento di una tabella, eseguire una query sulla vista del catalogo di sistema SVV_TABLE_INFO. Per ulteriori informazioni, consulta SVV_TABLE_INFO. Per visualizzare i suggerimenti di Amazon Redshift Advisor per le tabelle, eseguire una query sulla vista del catalogo di sistema SVV_ALTER_TABLE_RECOMMENDATIONS. Per ulteriori informazioni, consulta SVV_ALTER_TABLE_RECOMMENDATIONS. Per visualizzare le azioni intraprese da Amazon Redshift, eseguire una query sulla vista del catalogo di sistema SVL_AUTO_WORKER_ACTION. Per ulteriori informazioni, consulta SVL_AUTO_WORKER_ACTION.

COMPOUND

Specifica che i dati sono ordinati utilizzando una chiave composta costituita da tutte le colonne elencate, nell'ordine in cui sono elencate. Una chiave di ordinamento composta è più utile quando una query esegue la scansione delle righe in base all'ordine delle colonne di ordinamento. I vantaggi in termini di prestazioni dell'ordinamento con una chiave composta diminuiscono quando le query si basano su colonne di ordinamento secondarie. Puoi definire un massimo di 400 colonne COMPOUND SORTKEY per tabella.

INTERLEAVED

Specifica che i dati sono ordinati utilizzando una chiave di ordinamento "interlacciato". È possibile specificare un massimo di otto colonne per una chiave di ordinamento "interlacciato".

L'ordinamento "interlacciato" fornisce uguale peso a ciascuna colonna, o sottoinsieme di colonne, nella chiave di ordinamento, quindi le query non dipendono dall'ordine delle colonne nella chiave di ordinamento. Quando una query utilizza una o più colonne di ordinamento secondarie, l'ordinamento "interlacciato" migliora in modo significativo le prestazioni delle

query. L'ordinamento "interlacciato" comporta un piccolo costo generale per il caricamento dei dati e le operazioni di vacuum.

> **Important**
>
> Non utilizzare una chiave di ordinamento "interlacciato" su colonne con attributi monotonicamente crescenti, come colonne di identità, date o timestamp.

ENCODE AUTO

Consente ad Amazon Redshift di regolare automaticamente il tipo di codifica per tutte le colonne della tabella per ottimizzare le prestazioni delle query. ENCODE AUTO conserva i tipi di codifica iniziali specificati durante la creazione della tabella. Quindi, se Amazon Redshift determina che un nuovo tipo di codifica può migliorare le prestazioni delle query, Amazon Redshift può modificare il tipo di codifica delle colonne della tabella. ENCODE AUTO è l'impostazione di default se non si specifica un tipo di codifica in nessuna colonna della tabella.

UNIQUE (column_name [,...])

Vincolo che specifica che un gruppo di una o più colonne di una tabella può contenere solo valori univoci. Il comportamento del vincolo di tabella univoco è uguale a quello dei vincoli di colonna, con la possibilità di estendersi su più colonne. Nel contesto dei vincoli univoci, i valori null non sono considerati uguali. Ogni vincolo di tabella univoco deve denominare un insieme di colonne diverso dal set di colonne denominato da qualsiasi altro vincolo di chiave univoco o primario definito per la tabella.

> **Important**
>
> I vincoli univoci sono informativi e non vengono applicati dal sistema.

PRIMARY KEY (column_name [,...])

Vincolo che specifica che una colonna o un numero di colonne di una tabella può contenere solo valori non null univoci (non duplicati). L'identificazione di un set di colonne come chiave primaria fornisce anche i metadati relativi alla progettazione dello schema. Una chiave primaria implica che altre tabelle possono fare affidamento su questo set di colonne come identificatore univoco per le righe. È possibile specificare una chiave primaria per una tabella, sia come vincolo di una singola

colonna sia come vincolo di tabella. Il vincolo di chiave primaria deve nominare un set di colonne diverso da altri set di colonne nominati da qualsiasi vincolo univoco definito per la stessa tabella.

> **Important**
>
> I vincoli della chiave primaria sono solo informativi. Non vengono applicati dal sistema, ma vengono utilizzati dal pianificatore.

FOREIGN KEY (column_name [, ...]) REFERENCES reftable [(refcolumn)]

Vincolo che specifica un vincolo di chiave esterna, che richiede che un gruppo di una o più colonne della nuova tabella contenga solo valori che corrispondono a quelli nelle colonne di riferimento di alcune righe della tabella di riferimento. Se refcolumn viene omesso, viene usata la chiave primaria di reftable. Le colonne di riferimento devono essere le colonne di un vincolo di chiave primaria o univoco nella tabella di riferimento.

> **Important**
>
> I vincoli della chiave esterna sono solo informativi. Non vengono applicati dal sistema, ma vengono utilizzati dal pianificatore.

Note per l'utilizzo

In modo univoco, le limitazioni della chiave esterna e della chiave primaria sono solo a livello informativo e non vengono applicate da Amazon Redshift quando si popola una tabella. Ad esempio, se si inseriscono dati in una tabella con dipendenze, l'inserimento può avere esito positivo anche se viola il vincolo. Tuttavia, la chiave esterna e quella primaria vengono utilizzate come suggerimenti di pianificazione. Queste dovrebbero essere dichiarate se il processo ETL o altri processi nell'applicazione applicano la loro integrità. Per informazioni su come eliminare una tabella con dipendenze, consultare DROP TABLE.

Limiti e quote

Quando crei una tabella, considera i seguenti limiti.

- Esiste un limite per il numero massimo di tabelle in un cluster per tipo di nodo. Per ulteriori informazioni, consulta Limiti nella Guida alla gestione di Amazon Redshift.

- Il numero massimo di caratteri per un nome di tabella è 127.
- Il numero massimo di colonne che puoi definire in una singola tabella è 1.600.
- Il numero massimo di colonne SORTKEY che puoi definire in una singola tabella è 400.

Riepilogo delle impostazioni a livello di colonna e delle impostazioni a livello di tabella

Diversi attributi e impostazioni possono essere impostati a livello di colonna o a livello di tabella. In alcuni casi, l'impostazione di un attributo o vincolo a livello di colonna o a livello di tabella ha lo stesso effetto. In altri casi, producono risultati diversi.

Il seguente elenco riepiloga le impostazioni a livello di colonna e a livello di tabella:

DISTKEY

Non vi è alcuna differenza in termini di effetti se impostati a livello di colonna o a livello di tabella.

Se DISTKEY è impostato, a livello di colonna o a livello di tabella, DISTSTYLE deve essere impostato su KEY o non impostato affatto. DISTSTYLE può essere impostato solo a livello di tabella.

SORTKEY

Se impostato a livello di colonna, SORTKEY deve essere una singola colonna. Se SORTKEY è impostato a livello di tabella, una o più colonne possono costituire una chiave di ordinamento composto o "interlacciato" composto.

COLLATE CASE_SENSITIVE | COLLATE CASE_INSENSITIVE

Amazon Redshift non supporta la modifica della configurazione della distinzione tra maiuscole e minuscole per una colonna. Quando si aggiunge una nuova colonna alla tabella, Amazon Redshift utilizza il valore predefinito per la distinzione tra maiuscole e minuscole. Amazon Redshift non supporta la parola chiave COLLATE quando si aggiunge una nuova colonna.

Per informazioni su come creare i database mediante il confronto di database, consultare CREATE DATABASE.

Per ulteriori informazioni sulla funzione COLLATE, consultare Funzione COLLATE.

UNIQUE

A livello di colonna, una o più chiavi possono essere impostate su UNIQUE; il vincolo UNIQUE si applica a ciascuna colonna singolarmente. Se UNIQUE è impostato a livello di tabella, una o più colonne possono costituire un vincolo UNIQUE composito.

PRIMARY KEY

Se impostato a livello di colonna, PRIMARY KEY deve essere una singola colonna. Se PRIMARY KEY è impostato a livello di tabella, una o più colonne possono costituire una chiave primaria.

FOREIGN KEY

Non vi è alcuna differenza in termini di effetti se FOREIGN KEY è impostato a livello di colonna o a livello di tabella. A livello di colonna, la sintassi è semplicemente `REFERENCES` reftable [(refcolumn)].

Distribuzione dei dati in arrivo

Quando lo schema di distribuzione dell'hash dei dati in arrivo corrisponde a quello della tabella di destinazione, non è necessaria alcuna distribuzione fisica dei dati quando i dati vengono caricati. Ad esempio, se una chiave di distribuzione è impostata per la nuova tabella e i dati vengono inseriti da un'altra tabella che viene distribuita sulla stessa colonna chiave, i dati vengono caricati in posizione, utilizzando gli stessi nodi e sezioni. Tuttavia, se le tabelle di origine e di destinazione sono entrambe impostate sulla distribuzione EVEN, i dati vengono ridistribuiti nella tabella di destinazione.

Tabelle estese

Potrebbe essere possibile creare una tabella molto ampia ma non essere in grado di eseguire l'elaborazione della query, come le istruzioni INSERT o SELECT, sulla tabella. La larghezza massima di una tabella con colonne a larghezza fissa, come CHAR, è 64 KB - 1 (o 65535 byte). Se la tabella include le colonne VARCHAR, può avere una larghezza dichiarata più ampia senza restituire un errore poiché le colonne VARCHARS non contribuiscono con la loro larghezza dichiarata completa al limite calcolato di elaborazione della query. Il limite effettivo di elaborazione della query con le colonne VARCHAR varierà in base a una serie di fattori.

Se una tabella è troppo ampia per l'inserimento o la selezione, ricevi il seguente errore.

```
ERROR:  8001
DETAIL:  The combined length of columns processed in the SQL statement
exceeded the query-processing limit of 65535 characters (pid:7627)
```

Esempi

Per esempi che mostrano come utilizzare il comando CREATE TABLE, consulta l'argomento Esempi.

Esempi

I seguenti esempi dimostrano vari attributi di colonna e tabella nelle istruzioni CREATE TABLE di Amazon Redshift. Per ulteriori informazioni su CREATE TABLE, incluse le definizioni dei parametri, consulta CREATE TABLE.

Molti esempi utilizzano tabelle e dati del set di dati di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

È possibile aggiungere un prefisso al nome della tabella con il nome del database e il nome dello schema in un comando CREATE TABLE. Ad esempio, `dev_database.public.sales`. Il nome del database deve essere il database a cui si è connessi. Qualsiasi tentativo di creare oggetti di database in un altro database restituisce un errore di operazione non valida.

Creazione di una tabella con una chiave di distribuzione, una chiave di ordinamento composta e una compressione

L'esempio seguente crea una tabella SALES nel database TICKIT con compressione definita per più colonne. LISTID è dichiarato come chiave di distribuzione e LISTID e SELLERID sono dichiarati come una chiave di ordinamento composto a più colonne. Anche i vincoli della chiave primaria e della chiave esterna sono definiti per la tabella. Prima di creare la tabella nell'esempio, potrebbe essere necessario aggiungere un vincolo UNIQUE a ogni colonna a cui fa riferimento una chiave esterna, se i vincoli non esistono.

```
create table sales(
salesid integer not null,
listid integer not null,
sellerid integer not null,
buyerid integer not null,
eventid integer not null encode mostly16,
dateid smallint not null,
qtysold smallint not null encode mostly8,
pricepaid decimal(8,2) encode delta32k,
commission decimal(8,2) encode delta32k,
saletime timestamp,
primary key(salesid),
foreign key(listid) references listing(listid),
foreign key(sellerid) references users(userid),
foreign key(buyerid) references users(userid),
foreign key(dateid) references date(dateid))
distkey(listid)
```

```
compound sortkey(listid,sellerid);
```

I risultati sono i seguenti:

```
schemaname | tablename | column     | type                          | encoding | distkey
 | sortkey | notnull
-----------+-----------+------------+-------------------------------+----------+---------
+---------+--------
public     | sales     | salesid    | integer                       | lzo      | false
 |       0 | true
public     | sales     | listid     | integer                       | none     | true
 |       1 | true
public     | sales     | sellerid   | integer                       | none     | false
 |       2 | true
public     | sales     | buyerid    | integer                       | lzo      | false
 |       0 | true
public     | sales     | eventid    | integer                       | mostly16 | false
 |       0 | true
public     | sales     | dateid     | smallint                      | lzo      | false
 |       0 | true
public     | sales     | qtysold    | smallint                      | mostly8  | false
 |       0 | true
public     | sales     | pricepaid  | numeric(8,2)                  | delta32k | false
 |       0 | false
public     | sales     | commission | numeric(8,2)                  | delta32k | false
 |       0 | false
public     | sales     | saletime   | timestamp without time zone | lzo      | false
 |       0 | false
```

Nell'esempio seguente viene creata la tabella t1 con una colonna col1 senza distinzione tra maiuscole e minuscole.

```
create table T1 (
  col1 Varchar(20) collate case_insensitive
 );

insert into T1 values ('bob'), ('john'), ('Tom'), ('JOHN'), ('Bob');
```

Esecuzione di query sulla tabella:

```
select * from T1 where col1 = 'John';
```

```
col1
------
 john
 JOHN
(2 rows)
```

Creazione di una tabella usando una chiave di ordinamento interlacciato

L'esempio seguente crea la tabella CUSTOMER con una chiave di ordinamento "interlacciato".

```
create table customer_interleaved (
  c_custkey     integer        not null,
  c_name        varchar(25)    not null,
  c_address     varchar(25)    not null,
  c_city        varchar(10)    not null,
  c_nation      varchar(15)    not null,
  c_region      varchar(12)    not null,
  c_phone       varchar(15)    not null,
  c_mktsegment      varchar(10)    not null)
diststyle all
interleaved sortkey (c_custkey, c_city, c_mktsegment);
```

Creazione di una tabella utilizzando IF NOT EXISTS

L'esempio seguente crea la tabella CITIES oppure non esegue nulla e restituisce un messaggio se lo schema esiste già:

```
create table if not exists cities(
cityid integer not null,
city varchar(100) not null,
state char(2) not null);
```

Creazione di una tabella con distribuzione ALL

L'esempio seguente crea la tabella VENUE con la distribuzione ALL.

```
create table venue(
venueid smallint not null,
venuename varchar(100),
venuecity varchar(30),
venuestate char(2),
venueseats integer,
```

```
primary key(venueid))
diststyle all;
```

Creazione di una tabella con distribuzione EVEN

Il seguente esempio crea una tabella chiamata MYEVENT con tre colonne.

```
create table myevent(
eventid int,
eventname varchar(200),
eventcity varchar(30))
diststyle even;
```

La tabella è distribuita in modo uniforme e non è ordinata. La tabella non ha colonne DISTKEY o SORTKEY dichiarate.

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'myevent';

  column   |          type          | encoding | distkey | sortkey
-----------+------------------------+----------+---------+---------
 eventid   | integer                | lzo      | f       |       0
 eventname | character varying(200) | lzo      | f       |       0
 eventcity | character varying(30)  | lzo      | f       |       0
(3 rows)
```

Creazione di una tabella temporanea simile a un'altra tabella

Nell'esempio seguente viene creata una tabella temporanea denominata TEMPEVENT che eredita le colonne dalla tabella EVENT.

```
create temp table tempevent(like event);
```

Questa tabella eredita anche gli attributi DISTKEY e SORTKEY della tabella padre:

```
select "column", type, encoding, distkey, sortkey
 from pg_table_def where tablename = 'tempevent';

  column   |            type             | encoding | distkey | sortkey
-----------+-----------------------------+----------+---------+---------
 eventid   | integer                     | none     | t       |       1
```

```
 venueid   | smallint                    | none     | f       |       0
 catid     | smallint                    | none     | f       |       0
 dateid    | smallint                    | none     | f       |       0
 eventname | character varying(200)      | lzo      | f       |       0
 starttime | timestamp without time zone | bytedict | f       |       0
(6 rows)
```

Creazione di una tabella con una colonna IDENTITY

Nell'esempio seguente viene creata una tabella denominata VENUE_IDENT che ha una colonna IDENTITY denominata VENUEID. Questa colonna inizia con 0 e incrementa di 1 per ogni record. VENUEID è anche dichiarato come chiave primaria della tabella.

```
create table venue_ident(venueid bigint identity(0, 1),
venuename varchar(100),
venuecity varchar(30),
venuestate char(2),
venueseats integer,
primary key(venueid));
```

Creazione di una tabella con una colonna IDENTITY predefinita

L'esempio seguente crea una tabella denominata `t1`. Questa tabella contiene una colonna IDENTITY denominata `hist_id` e una colonna IDENTITY predefinita denominata `base_id`.

```
CREATE TABLE t1(
  hist_id BIGINT IDENTITY NOT NULL, /* Cannot be overridden */
  base_id BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, /* Can be overridden */
  business_key varchar(10) ,
  some_field varchar(10)
);
```

L'inserimento di una riga nella tabella mostra che entrambi i valori `hist_id` e `base_id` vengono generati.

```
INSERT INTO T1 (business_key, some_field) values ('A','MM');
```

```
SELECT * FROM t1;

 hist_id | base_id | business_key | some_field
```

```
---------+---------+--------------+------------
       1 |       1 | A            | MM
```

L'inserimento di una seconda riga mostra che viene generato il valore predefinito per `base_id`.

```
INSERT INTO T1 (base_id, business_key, some_field) values (DEFAULT, 'B','MNOP');
```

```
SELECT * FROM t1;

 hist_id | base_id | business_key | some_field
---------+---------+--------------+------------
       1 |       1 | A            | MM
       2 |       2 | B            | MNOP
```

L'inserimento di una terza riga mostra che il valore per `base_id` non deve essere univoco.

```
INSERT INTO T1 (base_id, business_key, some_field) values (2,'B','MNNN');
```

```
SELECT * FROM t1;

 hist_id | base_id | business_key | some_field
---------+---------+--------------+------------
       1 |       1 | A            | MM
       2 |       2 | B            | MNOP
       3 |       2 | B            | MNNN
```

Creazione di una tabella con valori di colonna DEFAULT

L'esempio seguente crea una tabella CATEGORYDEF che dichiara i valori predefiniti per ogni colonna:

```
create table categorydef(
catid smallint not null default 0,
catgroup varchar(10) default 'Special',
catname varchar(10) default 'Other',
catdesc varchar(50) default 'Special events',
primary key(catid));

insert into categorydef values(default,default,default,default);
```

```
select * from categorydef;

 catid | catgroup | catname |    catdesc
-------+----------+---------+----------------
     0 | Special  | Other   | Special events
(1 row)
```

Opzioni DISTSTYLE, DISTKEY e SORTKEY

L'esempio seguente mostra come funzionano le opzioni DISTKEY, SORTKEY e DISTSTYLE. In questo esempio, COL1 è la chiave di distribuzione e pertanto, lo stile di distribuzione deve essere impostato su KEY o non impostato. Per impostazione predefinita, la tabella non dispone della chiave di ordinamento e quindi non è ordinata:

```
create table t1(col1 int distkey, col2 int) diststyle key;
```

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 't1';

column |  type   | encoding | distkey | sortkey
-------+---------+----------+---------+---------
col1   | integer | az64     | t       | 0
col2   | integer | az64     | f       | 0
```

Nell'esempio seguente, la stessa colonna è definita come la chiave di distribuzione e la chiave di ordinamento. Di nuovo, lo stile di distribuzione deve essere impostato su KEY o non impostato.

```
create table t2(col1 int distkey sortkey, col2 int);
```

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 't2';

column |  type   | encoding | distkey | sortkey
-------+---------+----------+---------+---------
col1   | integer | none     | t       | 1
col2   | integer | az64     | f       | 0
```

Nell'esempio seguente, nessuna colonna è impostata come chiave di distribuzione, COL2 è impostato come chiave di ordinamento e lo stile di distribuzione è impostato su ALL:

```
create table t3(col1 int, col2 int sortkey) diststyle all;
```

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 't3';

Column |  Type   | Encoding | DistKey | SortKey
-------+---------+----------+---------+--------
col1   | integer | az64     | f       | 0
col2   | integer | none     | f       | 1
```

Nell'esempio seguente, lo stile di distribuzione è impostato su EVEN e nessuna chiave di ordinamento è definita esplicitamente, quindi la tabella è distribuita in modo uniforme ma non è ordinata.

```
create table t4(col1 int, col2 int) diststyle even;
```

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 't4';

             column |  type   |encoding | distkey | sortkey
--------+---------+---------+---------+--------
col1    | integer | az64    | f       | 0
col2    | integer | az64    | f       | 0
```

Creazione di una tabella con l'opzione ENCODE AUTO

L'esempio seguente crea la tabella `t1` con codifica di compressione automatica. ENCODE AUTO è l'impostazione di default per le tabelle quando non si specifica un tipo di codifica in nessuna colonna.

```
create table t1(c0 int, c1 varchar);
```

L'esempio seguente crea la tabella `t2` con codifica di compressione automatica specificando ENCODE AUTO.

```
create table t2(c0 int, c1 varchar) encode auto;
```

L'esempio seguente crea la tabella `t3` con codifica di compressione automatica specificando ENCODE AUTO. La colonna `c0` è definita con un tipo di codifica iniziale di DELTA. Amazon Redshift può modificare la codifica se un'altra codifica fornisce migliori prestazioni delle query.

```
create table t3(c0 int encode delta, c1 varchar) encode auto;
```

L'esempio seguente crea la tabella `t4` con codifica di compressione automatica specificando ENCODE AUTO. La colonna `c0` è definita con una codifica iniziale di DELTA e la colonna `c1` è definita con una codifica iniziale di LZO. Amazon Redshift può modificare queste codifiche se altre codifiche forniscono prestazioni di query migliori.

```
create table t4(c0 int encode delta, c1 varchar encode lzo) encode auto;
```

CREATE TABLE AS

Argomenti

Crea una nuova tabella in base a una query. Il proprietario di questa tabella è l'utente che invia il comando.

La nuova tabella viene caricata con i dati definiti dalla query nel comando. Le colonne della tabella hanno nomi e tipi di dati associati alle colonne di output della query. Il comando CREATE TABLE AS (CTAS) crea una nuova tabella e valuta la query per caricare la nuova tabella.

Sintassi

```
CREATE [ [ LOCAL ] { TEMPORARY | TEMP } ]
TABLE table_name
[ ( column_name [, ... ] ) ]
[ BACKUP { YES | NO } ]
[ table_attributes ]
AS query

where table_attributes are:
[ DISTSTYLE { AUTO | EVEN | ALL | KEY } ]
[ DISTKEY( distkey_identifier ) ]
[ [ COMPOUND | INTERLEAVED ] SORTKEY( column_name [, ...] ) ]
```

Parametri

LOCAL

Sebbene questa parola chiave opzionale sia accettata nell'istruzione, non ha alcun effetto in Amazon Redshift.

TEMPORARY | TEMP

Crea una tabella temporanea. La tabella temporanea viene automaticamente eliminata alla fine della sessione in cui è stata creata.

table_name

Nome della tabella da creare.

> **Important**
>
> Se specifichi un nome di tabella che inizia con "#", la tabella viene creata come tabella temporanea. Ad esempio:
>
> ```
> create table #newtable (id) as select * from oldtable;
> ```

La lunghezza massima per il nome della tabella è 127 byte; i nomi più lunghi vengono troncati a 127 byte. Amazon Redshift applica una quota del numero di tabelle per cluster per tipo di nodo. Il nome della tabella può essere qualificato con il nome del database e dello schema, come mostra la tabella seguente.

```
create table tickit.public.test (c1) as select * from oldtable;
```

Nell'esempio seguente, `tickit` è il nome del database e `public` è il nome dello schema. Se il database o lo schema non esiste, l'istruzione restituisce un errore.

Se viene specificato un nome dello schema, la nuova tabella viene creata in quello schema (presupponendo che il creatore abbia accesso allo schema). Il nome della tabella deve essere un nome univoco per lo schema. Se non viene specificato alcuno schema, la tabella viene creata utilizzando lo schema del database corrente. Se stai creando una tabella temporanea, non puoi specificare un nome schema perché le tabelle temporanee esistono in uno schema speciale.

Più tabelle temporanee con lo stesso nome possono esistere contemporaneamente nello stesso database se vengono create in sessioni separate. Queste tabelle sono assegnate a schemi diversi.

column_name

Nome di una colonna nella nuova tabella. Se non vengono forniti nomi di colonne, i nomi delle colonne vengono presi dai nomi delle colonne di output della query. I nomi di colonna predefiniti vengono utilizzati per le espressioni. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

BACKUP { YES | NO }

Clausola che specifica se la tabella deve essere inclusa negli snapshot di cluster automatizzati e manuali. Per le tabelle come le tabelle di gestione temporanea che non contengono dati critici, specificare BACKUP NO per risparmiare tempo di elaborazione durante la creazione di snapshot e il ripristino da snapshot e per ridurre lo spazio di archiviazione in Amazon Simple Storage Service. L'impostazione BACKUP NO non ha alcun effetto sulla replica automatica dei dati in altri nodi all'interno del cluster, pertanto le tabelle con BACKUP NO specificato vengono ripristinate in caso di errore del nodo. L'impostazione predefinita è BACKUP YES.

DISTSTYLE { AUTO | EVEN | KEY | ALL }

Definisce lo stile di distribuzione dei dati per l'intera tabella. Amazon Redshift distribuisce le righe di una tabella ai nodi di calcolo in base allo stile di distribuzione specificato per la tabella. Il valore di default è DISTSTYLE AUTO.

Lo stile di distribuzione selezionato per le tabelle influisce sulle prestazioni generali del database. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati.

- AUTO: Amazon Redshift assegna uno stile di distribuzione ottimale basato sui dati della tabella. Per visualizzare lo stile di distribuzione applicato a una tabella, eseguire una query sulla tabella del catalogo di sistema PG_CLASS. Per ulteriori informazioni, consulta Visualizzazione degli stili di distribuzione.
- EVEN: i dati della tabella sono distribuiti uniformemente tra i nodi di un cluster in una distribuzione round robin. Gli ID riga vengono utilizzati per determinare la distribuzione e approssimativamente lo stesso numero di righe viene distribuito a ciascun nodo. Questo è metodo di distribuzione predefinito.
- KEY: i dati sono distribuiti dai valori nella colonna DISTKEY. Quando imposti le colonne di unione delle tabelle di unione come chiavi di distribuzione, le righe di unione di entrambe le

tabelle sono collocate nei nodi di calcolo. Quando i dati sono collocati, l'ottimizzatore può eseguire i join in modo più efficiente. Se specifichi DISTSTYLE KEY, devi nominare una colonna DISTKEY.

- ALL: copia dell'intera tabella viene distribuita a ogni nodo. Questo stile di distribuzione garantisce che tutte le righe richieste per ogni join siano disponibili su ogni nodo, ma moltiplica i requisiti di storage e aumenta i tempi di caricamento e manutenzione per la tabella. La distribuzione ALL può migliorare i tempi di esecuzione quando viene utilizzata con determinate tabelle di dimensioni in cui la distribuzione KEY non è appropriata, ma i miglioramenti delle prestazioni devono essere confrontati con i costi di manutenzione.

DISTKEY (column)

Specifica un nome di colonna o un numero di posizione per la chiave di distribuzione. Utilizza il nome specificato nell'elenco delle colonne facoltativo per la tabella o l'elenco di selezione della query. In alternativa, utilizzare un numero di posizione, in cui la prima colonna selezionata è 1, la seconda è 2 e così via. Solo una colonna in una tabella può essere la chiave di distribuzione:

- Se dichiari una colonna come colonna DISTKEY, DISTSTYLE deve essere impostato su KEY o non impostato affatto.
- Se non dichiari una colonna DISTKEY, puoi impostare DISTSTYLE su EVEN.
- Se non specifichi DISTKEY o DISTSTYLE, CTAS determina lo stile di distribuzione per la nuova tabella in base al piano di query per la clausola SELECT. Per ulteriori informazioni, consulta Ereditarietà di attributi di colonna e tabella.

Puoi definire la stessa colonna della chiave di distribuzione e della chiave di ordinamento. Questo approccio tende ad accelerare i join quando la colonna in questione è una colonna di join nella query.

[COMPOUND | INTERLEAVED] SORTKEY (column_name [, ...])

Specifica una o più chiavi di ordinamento per la tabella. Quando vengono caricati nella tabella, i dati vengono ordinati in base alle colonne designate come chiavi di ordinamento.

È possibile specificare facoltativamente lo stile di ordinamento COMPOUND o INTERLEAVED. L'impostazione predefinita è COMPOUND. Per ulteriori informazioni, consulta Utilizzo delle chiavi di ordinamento.

È possibile definire un massimo di 400 colonne COMPOUND SORTKEY o 8 colonne INTERLEAVED SORTKEY per tabella.

Se non specifichi SORTKEY, CTAS determina le chiavi di ordinamento per la nuova tabella in base al piano di query per la clausola SELECT. Per ulteriori informazioni, consulta Ereditarietà di attributi di colonna e tabella.

COMPOUND

Specifica che i dati sono ordinati utilizzando una chiave composta costituita da tutte le colonne elencate, nell'ordine in cui sono elencate. Una chiave di ordinamento composta è più utile quando una query esegue la scansione delle righe in base all'ordine delle colonne di ordinamento. I vantaggi in termini di prestazioni dell'ordinamento con una chiave composta diminuiscono quando le query si basano su colonne di ordinamento secondarie. Puoi definire un massimo di 400 colonne COMPOUND SORTKEY per tabella.

INTERLEAVED

Specifica che i dati sono ordinati utilizzando una chiave di ordinamento "interlacciato". È possibile specificare un massimo di otto colonne per una chiave di ordinamento "interlacciato".

L'ordinamento "interlacciato" fornisce uguale peso a ciascuna colonna, o sottoinsieme di colonne, nella chiave di ordinamento, quindi le query non dipendono dall'ordine delle colonne nella chiave di ordinamento. Quando una query utilizza una o più colonne di ordinamento secondarie, l'ordinamento "interlacciato" migliora in modo significativo le prestazioni delle query. L'ordinamento "interlacciato" comporta un piccolo costo generale per il caricamento dei dati e le operazioni di vacuum.

AS query

Qualsiasi query (istruzione SELECT) supportata da Amazon Redshift.

Note per l'utilizzo di CTAS

Limiti

Amazon Redshift applica una quota del numero di tabelle per cluster per tipo di nodo.

Il numero massimo di caratteri per un nome di tabella è 127.

Il numero massimo di colonne che puoi definire in una singola tabella è 1.600.

Ereditarietà di attributi di colonna e tabella

Le tabelle CREATE TABLE AS (CTAS) non ereditano vincoli, colonne di identità, valori di colonna predefiniti o la chiave primaria dalla tabella da cui sono stati creati.

Non è possibile specificare le codifiche di compressione delle colonne per le tabelle CTAS. Amazon Redshift assegna automaticamente la codifica della compressione come segue:

- Le colonne definite come chiavi di ordinamento vengono assegnate alla compressione RAW.
- Alle colonne definite come tipi di dati BOOLEAN, REAL, DOUBLE PRECISION, GEOMETRY o GEOGRAPHY viene assegnata la compressione di tipo RAW.
- Le colonne definite come SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIME, TIMETZ, TIMESTAMP o TIMESTAMPTZ sono assegnate alla compressione AZ64.
- Le colonne definite come CHAR, VARCHAR o VARBYTE sono assegnate alla compressione LZO.

Per ulteriori informazioni, consulta Codifiche di compressione e Tipi di dati.

Per assegnare esplicitamente le codifiche di colonna, utilizza CREATE TABLE.

CTAS determina lo stile di distribuzione e la chiave di ordinamento per la nuova tabella in base al piano di query per la clausola SELECT.

Per le query complesse, come le query che includono join, aggregazioni, una clausola order by o una clausola limit, CTAS tenta di scegliere al meglio lo stile di distribuzione ottimale e la chiave di ordinamento in base al piano di query.

> **Note**
>
> Per le prestazioni ottimali con set di dati di grandi dimensioni o query complesse, consigliamo di eseguire il test utilizzando set di dati tipici.

Spesso puoi prevedere quale chiave di distribuzione e chiave di ordinamento viene scelta da CTAS esaminando il piano di query per vedere quali colonne, se presenti, l'ottimizzatore di query sceglie per l'ordinamento e la distribuzione dei dati. Se il nodo superiore del piano di query è una semplice scansione sequenziale da una singola tabella (XN Seq Scan), in genere CTAS utilizza lo stile di distribuzione e la chiave di ordinamento della tabella di origine. Se il nodo principale del piano di query è qualcosa di diverso da una scansione sequenziale (come XN Limit, XN Sort, XN e così via) HashAggregate, CTAS fa del suo meglio per scegliere lo stile di distribuzione e la chiave di ordinamento ottimali in base al piano di query.

Ad esempio, supponiamo di creare cinque tabelle utilizzando i seguenti tipi di clausole SELECT:

- Una semplice istruzione select
- Una clausola limit
- Una clausola order by usando LISTID
- Una clausola order by usando QTYSOLD
- Una funzione di aggregazione SUM con una clausola group by.

I seguenti esempi mostrano il piano di query per ciascuna istruzione CTAS.

```
explain create table sales1_simple as select listid, dateid, qtysold from sales;
                          QUERY PLAN
-------------------------------------------------------------
 XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=8)
(1 row)


explain create table sales2_limit as select listid, dateid, qtysold from sales limit
 100;
                             QUERY PLAN
---------------------------------------------------------------------
 XN Limit  (cost=0.00..1.00 rows=100 width=8)
   ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=8)
(2 rows)


explain create table sales3_orderbylistid as select listid, dateid, qtysold from sales
 order by listid;
                              QUERY PLAN
---------------------------------------------------------------------------
 XN Sort  (cost=1000000016724.67..1000000017155.81 rows=172456 width=8)
   Sort Key: listid
   ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=8)
(3 rows)


explain create table sales4_orderbyqty as select listid, dateid, qtysold from sales
 order by qtysold;
                              QUERY PLAN
---------------------------------------------------------------------------
 XN Sort  (cost=1000000016724.67..1000000017155.81 rows=172456 width=8)
   Sort Key: qtysold
   ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=8)
```

```
(3 rows)


explain create table sales5_groupby as select listid, dateid, sum(qtysold) from sales
 group by listid, dateid;
                            QUERY PLAN
------------------------------------------------------------------------
 XN HashAggregate  (cost=3017.98..3226.75 rows=83509 width=8)
   ->  XN Seq Scan on sales  (cost=0.00..1724.56 rows=172456 width=8)
(2 rows)
```

Per visualizzare la chiave di distribuzione e la chiave di ordinamento per ogni tabella, eseguire una query sulla tabella del catalogo di sistema PG_TABLE_DEF, come illustrato di seguito.

```
select * from pg_table_def where tablename like 'sales%';

      tablename       |   column   | distkey | sortkey
----------------------+------------+---------+---------
 sales                | salesid    | f       |       0
 sales                | listid     | t       |       0
 sales                | sellerid   | f       |       0
 sales                | buyerid    | f       |       0
 sales                | eventid    | f       |       0
 sales                | dateid     | f       |       1
 sales                | qtysold    | f       |       0
 sales                | pricepaid  | f       |       0
 sales                | commission | f       |       0
 sales                | saletime   | f       |       0
 sales1_simple        | listid     | t       |       0
 sales1_simple        | dateid     | f       |       1
 sales1_simple        | qtysold    | f       |       0
 sales2_limit         | listid     | f       |       0
 sales2_limit         | dateid     | f       |       0
 sales2_limit         | qtysold    | f       |       0
 sales3_orderbylistid | listid     | t       |       1
 sales3_orderbylistid | dateid     | f       |       0
 sales3_orderbylistid | qtysold    | f       |       0
 sales4_orderbyqty    | listid     | t       |       0
 sales4_orderbyqty    | dateid     | f       |       0
 sales4_orderbyqty    | qtysold    | f       |       1
 sales5_groupby       | listid     | f       |       0
 sales5_groupby       | dateid     | f       |       0
 sales5_groupby       | sum        | f       |       0
```

La tabella seguente riepiloga i risultati. Per semplicità, vengono omessi i dettagli relativi a costi, righe e larghezza dal piano explain.

Tabella	Istruzione CTAS SELECT	Nodo superiore del piano explain	Chiave distribuzione	Chiave di ordinamento
S1_SIMPLE	`select listid, dateid, qtysold from sales`	`XN Seq Scan on sales ...`	LISTID	DATEID
S2_LIMIT	`select listid, dateid, qtysold from sales limit 100`	`XN Limit ...`	Nessuna (EVEN)	Nessuno
S3_ORDER_BY_LISTID	`select listid, dateid, qtysold from sales order by listid`	`XN Sort ...` `Sort Key: listid`	LISTID	LISTID
S4_ORDER_BY_QTY	`select listid, dateid, qtysold from sales order by qtysold`	`XN Sort ...` `Sort Key: qtysold`	LISTID	QTYSOLD
S5_GROUP_BY	`select listid, dateid, sum(qtysold) from sales group by listid, dateid`	`XN HashAggregate ...`	Nessuna (EVEN)	Nessuno

Puoi specificare in modo lo stile di distribuzione e la chiave di ordinamento nell'istruzione CTAS. Ad esempio, la seguente istruzione crea una tabella utilizzando la distribuzione EVEN e specifica SALESID come chiave di ordinamento.

```
create table sales_disteven
diststyle even
sortkey (salesid)
as
select eventid, venueid, dateid, eventname
from event;
```

Codifica di compressione

ENCODE AUTO è l'impostazione predefinita per le tabelle. Amazon Redshift gestisce automaticamente la codifica di compressione per tutte le colonne della tabella.

Distribuzione dei dati in arrivo

Quando lo schema di distribuzione dell'hash dei dati in arrivo corrisponde a quello della tabella di destinazione, non è necessaria alcuna distribuzione fisica dei dati quando i dati vengono caricati. Ad esempio, se una chiave di distribuzione è impostata per la nuova tabella e i dati vengono inseriti da un'altra tabella che viene distribuita sulla stessa colonna chiave, i dati vengono caricati in posizione, utilizzando gli stessi nodi e sezioni. Tuttavia, se le tabelle di origine e di destinazione sono entrambe impostate sulla distribuzione EVEN, i dati vengono ridistribuiti nella tabella di destinazione.

Operazioni ANALYZE automatiche

Amazon Redshift analizza automaticamente le tabelle create con i comandi CTAS. Non è necessario eseguire il comando ANALYZE su queste tabelle quando vengono create per la prima volta. Se le modifichi, devi analizzarle come fai per le altre tabelle.

Esempi di CTAS

Il seguente esempio crea una tabella chiamata EVENT_BACKUP per la tabella EVENT:

```
create table event_backup as select * from event;
```

La tabella risultante eredita le chiavi di distribuzione e ordinamento dalla tabella EVENT.

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'event_backup';

column    | type                        | encoding | distkey | sortkey
----------+-----------------------------+----------+---------+--------
catid     | smallint                    | none     | false   |       0
```

```
dateid    | smallint                    | none     | false   |       1
eventid   | integer                     | none     | true    |       0
eventname | character varying(200)      | none     | false   |       0
starttime | timestamp without time zone | none     | false   |       0
venueid   | smallint                    | none     | false   |       0
```

Il seguente comando crea una nuova tabella chiamata EVENTDISTSORT selezionando quattro colonne dalla tabella EVENT. La nuova tabella viene distribuita da EVENTID e ordinata per EVENTID e DATEID:

```
create table eventdistsort
distkey (1)
sortkey (1,3)
as
select eventid, venueid, dateid, eventname
from event;
```

Il risultato è illustrato di seguito.

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'eventdistsort';

column   |          type          | encoding | distkey | sortkey
---------+------------------------+----------+---------+-------
eventid   | integer                | none     | t       | 1
venueid   | smallint               | none     | f       | 0
dateid    | smallint               | none     | f       | 2
eventname | character varying(200)| none     | f       | 0
```

Puoi creare esattamente la stessa tabella utilizzando i nomi delle colonne per le chiavi di distribuzione e ordinamento. Ad esempio:

```
create table eventdistsort1
distkey (eventid)
sortkey (eventid, dateid)
as
select eventid, venueid, dateid, eventname
from event;
```

La seguente istruzione applica la distribuzione uniforme alla tabella ma non definisce una chiave di ordinamento esplicita.

```
create table eventdisteven
diststyle even
as
select eventid, venueid, dateid, eventname
from event;
```

La tabella non eredita la chiave di ordinamento dalla tabella EVENT (EVENTID) perché la distribuzione EVEN è specificata per la nuova tabella. La nuova tabella non ha la chiave di ordinamento e la chiave di distribuzione.

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'eventdisteven';

column    |          type          | encoding | distkey | sortkey
----------+------------------------+----------+---------+---------
eventid   | integer                | none     | f       | 0
venueid   | smallint               | none     | f       | 0
dateid    | smallint               | none     | f       | 0
eventname | character varying(200) | none     | f       | 0
```

La seguente istruzione applica la distribuzione uniforme e definisce una chiave di ordinamento:

```
create table eventdistevensort diststyle even sortkey (venueid)
as select eventid, venueid, dateid, eventname from event;
```

La tabella risultante ha una chiave di ordinamento ma non una chiave di distribuzione.

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'eventdistevensort';

column    |          type          | encoding | distkey | sortkey
----------+------------------------+----------+---------+-------
eventid   | integer                | none     | f       | 0
venueid   | smallint               | none     | f       | 1
dateid    | smallint               | none     | f       | 0
eventname | character varying(200) | none     | f       | 0
```

La seguente istruzione ridistribuisce la tabella EVENT su una colonna chiave diversa dai dati in entrata che sono ordinati sulla colonna EVENTID e non definisce la colonna SORTKEY e quindi la tabella non è ordinata.

```
create table venuedistevent distkey(venueid)
as select * from event;
```

Il risultato è illustrato di seguito.

```
select "column", type, encoding, distkey, sortkey
from pg_table_def where tablename = 'venuedistevent';

 column   |            type             | encoding | distkey | sortkey
----------+-----------------------------+----------+---------+-------
eventid   | integer                     | none     | f       | 0
venueid   | smallint                    | none     | t       | 0
catid     | smallint                    | none     | f       | 0
dateid    | smallint                    | none     | f       | 0
eventname | character varying(200)      | none     | f       | 0
starttime | timestamp without time zone | none     | f       | 0
```

CREA UTENTE

Crea un nuovo utente del database. Gli utenti del database possono recuperare dati, eseguire comandi ed eseguire altre operazioni in un database, a seconda dei loro privilegi e ruoli. Per eseguire questo comando, è necessario essere un utente con privilegi avanzati del database.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per CREATE USER:

- Superuser
- Utenti con il privilegio CREATE USER

Sintassi

```
CREATE USER name [ WITH ]
PASSWORD { 'password' | 'md5hash' | 'sha256hash' | DISABLE }
[ option [ ... ] ]

where option can be:

CREATEDB | NOCREATEDB
```

```
| CREATEUSER | NOCREATEUSER
| SYSLOG ACCESS { RESTRICTED | UNRESTRICTED }
| IN GROUP groupname [, ... ]
| VALID UNTIL 'abstime'
| CONNECTION LIMIT { limit | UNLIMITED }
| SESSION TIMEOUT limit
| EXTERNALID external_id
```

Parametri

name

Il nome dell'utente da creare. Il nome utente non può essere PUBLIC. Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori.

WITH

Parola chiave facoltativa. WITH viene ignorato da Amazon Redshift

PASSWORD { 'password' | 'md5hash' | 'sha256hash' | DISABLE }

Imposta la password dell'utente.

Per impostazione predefinita, gli utenti possono modificare le proprie password, a meno che la password non sia disabilitata. Per disabilitare la password di un utente, specifica DISABLE. Quando la password di un utente è disabilitata, la password viene eliminata dal sistema e l'utente può accedere solo utilizzando credenziali utente temporanee AWS Identity and Access Management (IAM). Per ulteriori informazioni, consultare la sezione relativa all'utilizzo dell'autenticazione IAM per generare le credenziali dell'utente del database. Solo un utente con privilegi avanzati può abilitare o disabilitare le password. Non puoi disabilitare la password di un utente con privilegi avanzati. Per abilitare una password, esegui ALTER USER e specifica una password.

Puoi specificare la password in chiaro, come una stringa di hash MD5 o come una stringa di hash SHA256.

> **Note**
>
> Quando avvii un nuovo cluster utilizzando l'API o Amazon Redshift AWS Management Console AWS CLI, devi fornire una password di testo non crittografato per l'utente iniziale del database. Puoi modificare la password successivamente utilizzando ALTER USER.

Per il testo in chiaro, la password deve soddisfare i seguenti vincoli:

- La lunghezza deve essere compresa tra 8 e 64 caratteri.
- Deve contenere almeno una lettera maiuscola, una lettera minuscola e un numero.
- Può utilizzare qualsiasi carattere ASCII (codice ASCII da 33 a 126) tranne ' (virgolette singole), " (virgolette doppie), \, / o @.

Come alternativa più sicura al passaggio del parametro della password CREATE USER come testo in chiaro, puoi specificare un hash MD5 di una stringa che includa la password e il nome utente.

> **Note**
>
> Quando specifichi una stringa hash MD5, il comando CREATE USER verifica la presenza di una stringa hash MD5 valida, ma non convalida la porzione della password della stringa. In questo caso è possibile creare una password, ad esempio una stringa vuota, che non puoi utilizzare per accedere al database.

Per specificare una password MD5, attenersi alla seguente procedura:

1. Concatenare la password e il nome utente.

   Ad esempio, per la password `ez` e l'utente `user1`, la stringa concatenata è `ezuser1`.

2. Convertire la stringa concatenata in una stringa di hash MD5 a 32 caratteri. È possibile utilizzare qualsiasi utilità MD5 per creare la stringa hash. L'esempio seguente usa Funzione MD5 di Amazon Redshift e l'operatore di concatenazione (||) per restituire una stringa hash MD5 a 32 caratteri.

   ```
   select md5('ez' || 'user1');

   md5
   ----------------------------------
   153c434b4b77c89e6b94f12c5393af5b
   ```

3. Concatenare 'md5' davanti alla stringa di hash MD5 e fornire la stringa concatenata come argomento md5hash.

   ```
   create user user1 password 'md5153c434b4b77c89e6b94f12c5393af5b';
   ```

4. Accedi al database utilizzando le credenziali di accesso.

   Per questo esempio, accedere come `user1` con la password `ez`.

Un'altra alternativa sicura è quella di specificare un hash SHA-256 di una stringa di password; oppure è possibile fornire il proprio digest SHA-256 valido e il sale a 256 bit utilizzati per creare il digest.

- Digest — L'output di una funzione di hashing.
- Salt - Dati generati casualmente combinati con la password per ridurre i pattern nell'output della funzione di hashing.

```
'sha256|Mypassword'
```

```
'sha256|digest|256-bit-salt'
```

Nell'esempio seguente, Amazon Redshift genera e gestisce il salt.

```
CREATE USER admin PASSWORD 'sha256|Mypassword1';
```

Nell'esempio seguente vengono forniti un digest SHA-256 valido e un salt a 256 bit utilizzati per creare il digest.

Per specificare una password e impostarne l'hash con il tuo salt, segui questi passaggi:

1. Crea un salt a 256 bit. È possibile ottenere un salt utilizzando qualsiasi generatore di stringhe esadecimali che restituisce una stringa lunga 64 caratteri. In questo esempio, il salt è `c721bff5d9042cf541ff7b9d48fa8a6e545c19a763e3710151f9513038b0f6c6`.
2. Usa la funzione FROM_HEX per convertire il tuo salt in binario. Questa conversione è necessaria perché la funzione SHA2 richiede la rappresentazione binaria del salt. Vedi la seguente istruzione.

   ```
   SELECT
    FROM_HEX('c721bff5d9042cf541ff7b9d48fa8a6e545c19a763e3710151f9513038b0f6c6');
   ```

3. Usa la funzione CONCAT per aggiungere il salt alla password. In questo esempio, la password è `Mypassword1`. Vedi la seguente istruzione.

```
SELECT
 CONCAT('Mypassword1',FROM_HEX('c721bff5d9042cf541ff7b9d48fa8a6e545c19a763e3710151f951303
```

4. Usa la funzione SHA2 per creare un digest dalla combinazione di password e salt. Vedi la seguente istruzione.

```
SELECT
 SHA2(CONCAT('Mypassword1',FROM_HEX('c721bff5d9042cf541ff7b9d48fa8a6e545c19a763e3710151f95
 0);
```

5. Crea l'utente utilizzando il digest e il salt dei passaggi precedenti. Vedi la seguente istruzione.

```
CREATE USER admin PASSWORD 'sha256|
821708135fcc42eb3afda85286dee0ed15c2c461d000291609f77eb113073ec2|
c721bff5d9042cf541ff7b9d48fa8a6e545c19a763e3710151f9513038b0f6c6';
```

6. Accedi al database utilizzando le credenziali di accesso.

   Per questo esempio, accedere come `admin` con la password `Mypassword1`.

Se si imposta una password in testo normale senza specificare la funzione di hashing, viene generato un digest MD5 utilizzando il nome utente come salt.

CREATEDB | NOCREATEDB

L'opzione CREATEDB consente al nuovo utente di creare database. Il valore predefinito è NOCREATEDB.

CREATEUSER | NOCREATEUSER

L'opzione CREATEUSER crea un utente con privilegi avanzati con tutti i privilegi del database, incluso CREATE USER. L'impostazione predefinita è NOCREATEUSER. Per ulteriori informazioni, consulta superuser.

SYSLOG ACCESS { RESTRICTED | UNRESTRICTED }

Una clausola che specifica il livello di accesso che l'utente ha per tabelle e viste di sistema di Amazon Redshift.

Gli utenti normali che dispongono dell'autorizzazione SYSLOG ACCESS RESTRICTED possono vedere solo le righe generate da quell'utente nelle tabelle e nelle viste di sistema visibili all'utente. Il valore predefinito è RESTRICTED.

Gli utenti normali che dispongono dell'autorizzazione SYSLOG ACCESS UNRESTRICTED possono visualizzare tutte le righe nelle tabelle e nelle viste di sistema visibili all'utente, incluse le righe generate da un altro utente. UNRESTRICTED non fornisce un accesso regolare all'utente alle tabelle visibili agli utenti con privilegi avanzati. Solo gli utenti con privilegi avanzati possono vedere le tabelle visibili agli utenti con privilegi avanzati.

> **Note**
>
> Concedere all'utente un accesso illimitato alle tabelle di sistema offre all'utente la visibilità dei dati generati da altri utenti. Ad esempio, STL_QUERY e STL_QUERYTEXT contengono il testo completo delle istruzioni INSERT, UPDATE e DELETE, che potrebbero contenere dati sensibili generati dall'utente.

Tutte le righe in SVV_TRANSACTIONS sono visibili per tutti gli utenti.

Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

IN GROUP groupname

Specifica il nome di un gruppo esistente a cui appartiene l'utente. Possono essere elencati più nomi di gruppi.

VALID UNTIL abstime

L'opzione VALID UNTIL imposta un tempo assoluto dopo il quale la password dell'utente non è più valida. Per impostazione predefinita, la password non ha limiti di tempo.

CONNECTION LIMIT { limit | UNLIMITED }

Numero massimo di connessioni di database che l'utente può aprire contemporaneamente. Il limite non viene applicato per gli utente con privilegi avanzati. Utilizza la parola chiave UNLIMITED per consentire il numero massimo di connessioni simultanee. È possibile che venga applicato anche un limite al numero di connessioni per ciascun database. Per ulteriori informazioni, consulta CREATE DATABASE. Il valore predefinito è UNLIMITED. Per visualizzare le connessioni correnti, eseguire una query sulla vista di sistema STV_SESSIONS.

> **Note**
>
> Se si applicano entrambi i limiti di connessione utente e database, deve essere disponibile uno slot di connessione inutilizzato che rientra in entrambi i limiti quando un utente tenta di connettersi.

SESSION TIMEOUT limite

Il tempo massimo (in secondi) in cui una sessione rimane inattiva o inutilizzata. L'intervallo è compreso tra 60 secondi (un minuto) e 1.728.000 secondi (20 giorni). Se per l'utente non è impostato alcun timeout di sessione, verrà applicata l'impostazione del cluster. Per ulteriori informazioni, consulta Quote e limiti in Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Quando si imposta il timeout della sessione, viene applicato solo alle nuove sessioni.

Per visualizzare informazioni sulle sessioni utente attive, tra cui l'ora di inizio, il nome utente e il timeout della sessione, eseguire una query sulla vista di sistema STV_SESSIONS. Per visualizzare le informazioni sulla cronologia delle sessioni utente, eseguire una query sulla vista STL_SESSIONS. Per recuperare informazioni sugli utenti del database, inclusi i valori di timeout della sessione, eseguire una query sulla vista SVL_USER_INFO.

EXTERNALID external_id

L'identificativo dell'utente, associato a un provider di identità. L'utente deve avere la password disabilitata. Per ulteriori informazioni, consulta Native identity provider (IdP) federation for Amazon Redshift (Federazione di provider di identità nativi (IdP) per Amazon Redshift).

Note per l'utilizzo

Per impostazione predefinita, tutti gli utenti hanno i privilegi CREATE e USAGE nello schema PUBLIC. Per impedire agli utenti di creare oggetti nello schema PUBLIC di un database, utilizza il comando REVOKE per rimuovere tale privilegio.

Quando si utilizza l'autenticazione IAM per creare credenziali utente del database, puoi creare un utente con privilegi avanzati in grado di accedere solo utilizzando credenziali temporanee. Non puoi disabilitare la password di un utente con privilegi avanzati, ma puoi creare una password sconosciuta utilizzando una stringa di hash MD5 generata in modo casuale.

```
create user iam_superuser password 'md5A1234567890123456780123456789012' createuser;
```

Il caso di un nome utente racchiuso tra virgolette doppie è sempre conservato indipendentemente dall'impostazione dell'opzione di configurazione `enable_case_sensitive_identifier`. Per ulteriori informazioni, consulta enable_case_sensitive_identifier.

Esempi

Il seguente comando crea un utente denominato dbuser, con la password "abcD1234", i privilegi di creazione del database e un limite di connessione di 30.

```
create user dbuser with password 'abcD1234' createdb connection limit 30;
```

Eseguire una query sulla tabella di catalogo PG_USER_INFO per visualizzare i dettagli su un utente del database.

```
select * from pg_user_info;

 usename   | usesysid | usecreatedb | usesuper | usecatupd | passwd   | valuntil |
 useconfig | useconnlimit
-----------+----------+-------------+----------+-----------+----------+----------
+-----------+-------------
 rdsdb     |        1 | true        | true     | true      | ******** | infinity |
      |
 adminuser |      100 | true        | true     | false     | ******** |          |
      | UNLIMITED
 dbuser    |      102 | true        | false    | false     | ******** |          |
      | 30
```

Nell'esempio seguente, la password dell'account è valida fino al 10 giugno 2017.

```
create user dbuser with password 'abcD1234' valid until '2017-06-10';
```

L'esempio seguente crea un utente con una password che rispetta la distinzione tra lettere maiuscole e minuscole che contiene caratteri speciali.

```
create user newman with password '@AbC4321!';
```

Per utilizzare una barra rovesciata ('\') nella tua password MD5, specifica il carattere di escape per la barra rovesciata con una barra rovesciata nella stringa di origine. Nell'esempio seguente viene creato un utente denominato `slashpass` con una singola barra rovesciata ('\') come password.

```
select md5('\\'||'slashpass');

md5
--------------------------------
0c983d1a624280812631c5389e60d48c
```

Creazione di un utente con la password md5.

```
create user slashpass password 'md50c983d1a624280812631c5389e60d48c';
```

Nell'esempio seguente viene creato un utente denominato `dbuser` con un timeout di sessione inattiva impostato su 120 secondi.

```
CREATE USER dbuser password 'abcD1234' SESSION TIMEOUT 120;
```

L'esempio seguente crea un utente denominato bob. Lo spazio dei nomi è `myco_aad`. Si tratta solo di un esempio, Per eseguire correttamente il comando, è necessario disporre di un provider di identità registrato. Per ulteriori informazioni, consulta Native identity provider (IdP) federation for Amazon Redshift (Federazione di provider di identità nativi (IdP) per Amazon Redshift).

```
CREATE USER myco_aad:bob EXTERNALID "ABC123" PASSWORD DISABLE;
```

CREATE VIEW

Crea una vista in un database. La vista non è materializzata fisicamente, la query che definisce la vista viene eseguita ogni volta che vi si fa riferimento in una query. Per creare una vista con una tabella esterna, includi la clausola WITH NO SCHEMA BINDING.

Per creare una vista standard, devi accedere alle tabelle o alle viste sottostanti. Per eseguire una query su una vista standard, devi selezionare le autorizzazioni per la vista, ma non quelle per le tabelle sottostanti. Nel caso in cui crei una vista che fa riferimento a una tabella o a una vista in un altro schema o a una vista materializzata, potrebbero essere necessarie autorizzazioni aggiuntive. Per eseguire una query su una vista con associazione tardiva, devi selezionare le autorizzazioni

per la vista con associazione tardiva stessa. Inoltre devi assicurarti che il proprietario della vista con associazione tardiva abbia i privilegi di selezione per gli oggetti di riferimento (tabelle, viste o funzioni definite dall'utente). Per ulteriori informazioni sulle viste con associazione tardiva, consulta Note per l'utilizzo.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per CREATE VIEW:

- Per CREATE VIEW:
  - Superuser
  - Utenti con il privilegio CREATE [OR REPLACE] VIEW
- Per REPLACE VIEW:
  - Superuser
  - Utenti con il privilegio CREATE [OR REPLACE] VIEW
  - Proprietario della visualizzazione

Sintassi

```
CREATE [ OR REPLACE ] VIEW name [ ( column_name [, ...] ) ] AS query
[ WITH NO SCHEMA BINDING ]
```

Parametri

OR REPLACE

Se esiste già una vista con lo stesso nome, la vista viene sostituita. Puoi sostituire una vista solo con una nuova query che genera un set identico di colonne, usando gli stessi nomi di colonna e tipi di dati. CREATE OR REPLACE VIEW blocca le operazioni di lettura e scrittura della vista fino al completamento dell'operazione.

Quando una vista viene sostituita, vengono mantenute le altre proprietà, ad esempio la proprietà e i privilegi concessi.

name

Nome della vista. Se viene specificato un nome schema (come `myschema.myview`), la vista viene creata utilizzando lo schema specificato. Altrimenti, la vista viene creata nello schema

corrente. Il nome della vista deve essere diverso dal nome di qualsiasi altra vista o tabella nello stesso schema.

Se specifichi un nome di vista che inizia con "#", la vista viene creata come vista temporanea visibile solo nella sessione corrente.

Per ulteriori informazioni sui nomi validi, consultare Nomi e identificatori. Non è possibile creare tabelle o viste nei database di sistema template0, template1, padb_harvest o sys:internal.

column_name

Elenco facoltativo di nomi da utilizzare per le colonne nella vista. Se non vengono specificati nomi di colonne, i nomi delle colonne vengono ricavati dalla query. Il numero massimo di colonne che puoi definire in una singola vista è 1.600.

query

Query (sotto forma di istruzione SELECT) che valuta una tabella. Questa tabella definisce le colonne e le righe nella vista.

WITH NO SCHEMA BINDING

Clausola che specifica che la vista non è vincolata agli oggetti del database sottostante, come le tabelle e le funzioni definite dall'utente. Di conseguenza, non esiste alcuna dipendenza tra la vista e gli oggetti a cui fa riferimento. Puoi creare una vista anche se gli oggetti di riferimento non esistono. Poiché non esiste alcuna dipendenza, è possibile rimuovere o modificare un oggetto di riferimento senza influire sulla vista. Amazon Redshift non controlla le dipendenze finché non viene interrogata la vista. Per visualizzare i dettagli sulle viste con associazione tardiva, esegui la funzione PG_GET_LATE_BINDING_VIEW_COLS.

Quando includi la clausola WITH NO SCHEMA BINDING, le tabelle e le viste a cui si fa riferimento nell'istruzione SELECT devono essere qualificate con un nome schema. Lo schema deve esistere quando viene creata la vista, anche se la tabella di riferimento non esiste. Ad esempio, la seguente istruzione restituisce un errore.

```
create view myevent as select eventname from event
with no schema binding;
```

La seguente istruzione viene eseguita normalmente.

```
create view myevent as select eventname from public.event
with no schema binding;
```

> **Note**
>
> Non puoi eseguire operazioni di aggiornamento, inserimento o eliminazione da una vista.

Note per l'utilizzo

Viste con associazione tardiva

Una vista con associazione tardiva non controlla gli oggetti del database sottostante, come le tabelle e altre viste, finché non viene eseguita la query sulla vista. Di conseguenza, puoi modificare o rimuovere gli oggetti sottostanti senza rimuovere e ricreare la vista. Se rimuovi gli oggetti sottostanti, le query sulla vista con associazione tardiva avranno esito negativo. Se la query sulla vista con associazione tardiva fa riferimento a colonne nell'oggetto sottostante che non sono presenti, la query avrà esito negativo.

Se rimuovi e quindi ricrei una vista o una tabella sottostante di una vista con associazione tardiva, il nuovo oggetto viene creato con le autorizzazioni di accesso predefinite. Potrebbe essere necessario concedere le autorizzazioni agli oggetti sottostanti per gli utenti che eseguono query sulla vista.

Per creare una vista con associazione tardiva, includi la clausola WITH NO SCHEMA BINDING. Nell'esempio seguente viene creata una vista senza associazione di schema.

```
create view event_vw as select * from public.event
with no schema binding;
```

```
select * from event_vw limit 1;

eventid | venueid | catid | dateid | eventname     | starttime
--------+---------+-------+--------+---------------+--------------------
      2 |     306 |     8 |   2114 | Boris Godunov | 2008-10-15 20:00:00
```

L'esempio seguente mostra che puoi modificare una tabella sottostante senza ricreare la vista.

```
alter table event rename column eventname to title;
```

```
select * from event_vw limit 1;

eventid | venueid | catid | dateid | title         | starttime
```

```
--------+---------+-------+--------+---------------+--------------------
      2 |     306 |     8 |   2114 | Boris Godunov | 2008-10-15 20:00:00
```

È possibile fare riferimento alle tabelle esterne di Amazon Redshift Spectrum solo in una vista con associazione tardiva. Un'applicazione delle viste con associazione tardiva è quella di eseguire la query sulle tabelle Amazon Redshift e Redshift Spectrum. Ad esempio, è possibile utilizzare il comando UNLOAD per archiviare i dati meno recenti in Amazon S3. Quindi, creare una tabella esterna Redshift Spectrum che fa riferimento ai dati in Amazon S3 e una vista che esegue le query su entrambe le tabelle. Nell'esempio seguente viene utilizza una clausola UNION ALL per eseguire il join alla tabella `SALES` di Amazon Redshift e alla tabella `SPECTRUM.SALES` di Redshift Spectrum.

```
create view sales_vw as
select * from public.sales
union all
select * from spectrum.sales
with no schema binding;
```

Per ulteriori informazioni sulla creazione di tabelle esterne di Redshift Spectrum, inclusa la tabella `SPECTRUM.SALES`, vedi Nozioni di base su Amazon Redshift Spectrum.

Quando crei una vista standard da una vista con associazione tardiva, la definizione della vista standard contiene la definizione della vista con associazione tardiva. La dipendenza della vista con associazione tardiva non viene tracciata, quindi le modifiche alla vista con associazione tardiva non vengono tracciate nella vista standard.

Per aggiornare la vista standard in modo che faccia riferimento alla definizione più recente della vista con associazione tardiva, esegui CREATE OR REPLACE VIEW con la definizione iniziale utilizzata per creare la vista standard.

Di seguito è riportato un esempio di creazione di una vista standard da una vista con associazione tardiva.

```
create view sales_vw_lbv as
select * from public.sales
with no schema binding;

show view sales_vw_lbv;
                            Show View DDL statement
-------------------------------------------------------------------------------
 create view sales_vw_lbv as select * from public.sales with no schema binding;
(1 row)
```

```
create view sales_vw as
select * from sales_vw_lbv;

show view sales_vw;
                                                  Show View DDL statement
-------------------------------------------------------------------------------------------------------
 SELECT sales_vw_lbv.price, sales_vw_lbv."region" FROM (SELECT sales.price,
 sales."region" FROM sales) sales_vw_lbv;
(1 row)
```

Tieni presente che la vista con associazione tardiva, come illustrata nell'istruzione DDL per la vista standard, viene definita al momento della creazione della vista standard e non verrà aggiornata con le modifiche apportate successivamente alla vista con associazione tardiva.

Esempi

I comandi di esempio utilizzano un set di oggetti e dati di esempio chiamato database TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Il seguente comando crea una vista chiamata myevent da una tabella chiamata EVENT.

```
create view myevent as select eventname from event
where eventname = 'LeAnn Rimes';
```

Il seguente comando crea una vista chiamata myuser da una tabella chiamata USERS.

```
create view myuser as select lastname from users;
```

Il seguente comando crea o sostituisce una vista chiamata myuser da una tabella chiamata USERS.

```
create or replace view myuser as select lastname from users;
```

Nell'esempio seguente viene creata una vista senza associazione di schema.

```
create view myevent as select eventname from public.event
with no schema binding;
```

DEALLOCATE

Annulla l'allocazione di un'istruzione preparata.

Sintassi

```
DEALLOCATE [PREPARE] plan_name
```

Parametri

PREPARE

Questa parola chiave è facoltativa e viene ignorata.

plan_name

Il nome dell'istruzione preparata di cui annullare l'allocazione.

Note per l'utilizzo

DEALLOCATE viene utilizzato per annullare l'allocazione di un'istruzione SQL preparata in precedenza. Se non annulli esplicitamente l'allocazione di un'istruzione preparata, l'allocazione viene annullata al termine della sessione corrente. Per ulteriori informazioni sulle istruzioni preparate, consultare PREPARE.

Vedi anche

EXECUTE, PREPARE

DECLARE

Definisce un nuovo cursore. Utilizza un cursore per recuperare poche righe alla volta dal set di risultati di una query più grande.

Quando viene recuperata la prima riga di un cursore, l'intero set di risultati si materializza sul nodo principale, in memoria o sul disco, se necessario. A causa del potenziale impatto negativo sulle prestazioni dell'utilizzo dei cursori con set di risultati di grandi dimensioni, si consiglia di utilizzare approcci alternativi laddove possibile. Per ulteriori informazioni, consulta Considerazioni sulle prestazioni quando si utilizzano i cursori.

Devi dichiarare un cursore all'interno di un blocco di transazione. È possibile aprire un solo cursore alla volta per sessione.

Per ulteriori informazioni, consultare FETCH, CLOSE.

Sintassi

```
DECLARE cursor_name CURSOR FOR query
```

Parametri

cursor_name

Nome del nuovo cursore.

query

Istruzione SELECT che popola il cursore.

Note per l'utilizzo di DECLARE CURSOR

Se l'applicazione client utilizza una connessione ODBC e la query crea un set di risultati che è troppo grande per rientrare nella memoria, puoi eseguire il flusso del set di risultati all'applicazione client utilizzando un cursore. Quando usi un cursore, l'intero set di risultati si materializza sul nodo principale e il client può recuperare i risultati in modo incrementale.

> **Note**
>
> Per abilitare i cursori in ODBC per Microsoft Windows, abilitare l'opzione Usa dichiara/recupera nel DSN ODBC utilizzato per Amazon Redshift. Si consiglia di impostare la dimensione della cache ODBC, utilizzando il campo Cache Size (Dimensione cache) nella finestra di dialogo delle opzioni del DSN ODBC su 4.000 o un valore superiore sui cluster a più nodi per ridurre al minimo i round trip. Su un cluster a nodo singolo, imposta la dimensione della cache su 1.000.

A causa del potenziale impatto negativo sulle prestazioni dell'utilizzo dei cursori, si consiglia di utilizzare approcci alternativi laddove possibile. Per ulteriori informazioni, consulta Considerazioni sulle prestazioni quando si utilizzano i cursori.

I cursori Amazon Redshift sono supportati con le seguenti limitazioni:

- È possibile aprire un solo cursore alla volta per sessione.
- I cursori devono essere utilizzati all'interno di una transazione (BEGIN … END).

- La dimensione massima dei set di risultati cumulativi per tutti i cursori è vincolata in base al tipo di nodo del cluster. Se sono necessari set di risultati più grandi, puoi ridimensionare una configurazione di nodo XL o 8XL.

  Per ulteriori informazioni, consulta Vincoli del cursore.

Vincoli del cursore

Quando viene recuperata la prima riga di un cursore, l'intero set di risultati si materializza sul nodo principale. Se il set di risultati non rientra in memoria, viene scritto sul disco in base alle necessità. Per proteggere l'integrità del nodo principale, Amazon Redshift applica i vincoli sulla dimensione di tutti i set di risultati del cursore, in base al tipo di nodo del cluster.

La tabella seguente mostra le dimensioni massime del set di risultati totali per ciascun tipo di nodo del cluster. Le dimensioni massime del set di risultati sono espresse in megabyte.

Tipo di nodo	Dimensioni massime del set di risultati per cluster (MB)
Più nodi RA3 16XL	14400000
Nodo singolo DC2 Large	8000
Più nodi DC2 Large	192000
Più nodi DC2 8XL	3200000
Più nodi RA3 4XL	3200000
Nodi multipli RA3 XLPLUS	1000000
Nodo singolo RA3 XLPLUS	64000
Amazon Redshift Serverless	150000

Per visualizzare la configurazione del cursore attivo per un cluster, eseguire una query sulla tabella di sistema STV_CURSOR_CONFIGURATION come utente con privilegi avanzati. Per visualizzare lo stato dei cursori attivi, esegui la query sulla tabella di sistema STV_ACTIVE_CURSORS. Un utente può vedere solo le righe dei propri cursori, ma un utente con privilegi avanzati può visualizzare tutti i cursori.

Considerazioni sulle prestazioni quando si utilizzano i cursori

Poiché i cursori materializzano l'intero set di risultati sul nodo principale prima di iniziare a restituire i risultati al client, l'utilizzo di cursori con set di risultati molto grandi può avere un impatto negativo sulle prestazioni. Ti consigliamo pertanto di non utilizzare i cursori con set di risultati molto grandi. In alcuni casi, ad esempio quando l'applicazione utilizza una connessione ODBC, i cursori potrebbero essere l'unica soluzione attuabile. Se possibile, consigliamo di utilizzare queste alternative:

- Usa UNLOAD per esportare una tabella di grandi dimensioni. Quando si utilizza UNLOAD, i nodi di calcolo lavorano in parallelo per trasferire i dati direttamente nei file di dati in Amazon Simple Storage Service. Per ulteriori informazioni, consulta Scaricamento dei dati.
- Imposta il parametro relativo alle dimensioni di recupero JDBC nell'applicazione client. Se utilizzi una connessione JDBC e riscontri out-of-memory errori sul lato client, puoi consentire al client di recuperare i set di risultati in batch più piccoli impostando il parametro JDBC fetch size. Per ulteriori informazioni, consulta Impostazione del parametro delle dimensioni del recupero JDBC.

Esempi di DECLARE CURSOR

L'esempio seguente dichiara un cursore denominato LOLLAPALOOZA per selezionare le informazioni di vendita per l'evento Lollapalooza e quindi recupera le righe dal set di risultati utilizzando il cursore:

```
-- Begin a transaction

begin;

-- Declare a cursor

declare lollapalooza cursor for
select eventname, starttime, pricepaid/qtysold as costperticket, qtysold
from sales, event
where sales.eventid = event.eventid
and eventname='Lollapalooza';

-- Fetch the first 5 rows in the cursor lollapalooza:

fetch forward 5 from lollapalooza;

  eventname   |      starttime      | costperticket | qtysold
--------------+---------------------+---------------+---------
```

```
 Lollapalooza | 2008-05-01 19:00:00 |   92.00000000 |       3
 Lollapalooza | 2008-11-15 15:00:00 |  222.00000000 |       2
 Lollapalooza | 2008-04-17 15:00:00 |  239.00000000 |       3
 Lollapalooza | 2008-04-17 15:00:00 |  239.00000000 |       4
 Lollapalooza | 2008-04-17 15:00:00 |  239.00000000 |       1
(5 rows)

-- Fetch the next row:

fetch next from lollapalooza;

  eventname   |      starttime      | costperticket | qtysold
--------------+---------------------+---------------+---------
 Lollapalooza | 2008-10-06 14:00:00 |  114.00000000 |       2

-- Close the cursor and end the transaction:

close lollapalooza;
commit;
```

L'esempio seguente esegue un loop su un refcursor con tutti i risultati di una tabella:

```
CREATE TABLE tbl_1 (a int, b int);
INSERT INTO tbl_1 values (1, 2),(3, 4);

CREATE OR REPLACE PROCEDURE sp_cursor_loop() AS $$
DECLARE
    target record;
    curs1 cursor for select * from tbl_1;
BEGIN
    OPEN curs1;
    LOOP
        fetch curs1 into target;
        exit when not found;
        RAISE INFO 'a %', target.a;
    END LOOP;
    CLOSE curs1;
END;
$$ LANGUAGE plpgsql;

CALL sp_cursor_loop();

SELECT message
```

```
   from svl_stored_proc_messages
   where querytxt like 'CALL sp_cursor_loop()%';

  message
----------
      a 1
      a 3
```

DELETE

Elimina le righe dalle tabelle.

> **Note**
>
> Le dimensioni massime per una istruzione SQL è di 16 MB.

Sintassi

```
[ WITH [RECURSIVE] common_table_expression [, common_table_expression , ...] ]
DELETE [ FROM ] { table_name | materialized_view_name }
    [ { USING } table_name, ... ]
    [ WHERE condition ]
```

Parametri

Clausola WITH

Clausola facoltativa che specifica una o più common-table-expressions. Per informazioni, consultare Clausola WITH.

FROM

La parola chiave FROM è facoltativa eccetto quando la clausola USING è specificata. Le istruzioni `delete from event;` e `delete event;` sono operazioni equivalenti che rimuovono tutte le righe della tabella EVENT.

> **Note**
>
> Per eliminare tutte le righe da una tabella, usa il comando TRUNCATE sulla tabella. TRUNCATE è molto più efficiente di DELETE e non richiede VACUUM e ANALYZE.

Tuttavia, tieni presente che TRUNCATE esegue il commit della transazione in cui viene eseguito.

table_name

Tabella temporanea o persistente. Solo il proprietario della tabella o un utente con il privilegio DELETE sulla tabella può eliminare le righe dalla tabella.

Prendi in considerazione l'utilizzo del comando TRUNCATE per le operazioni di eliminazione rapida non qualificate su tabelle di grandi dimensioni; vedi TRUNCATE.

> **Note**
>
> Dopo aver eliminato un numero elevato di righe da una tabella:
>
> - Applica l'operazione di vacuum alla tabella per recuperare spazio di memorizzazione e riordinare le righe.
> - Analizza la tabella per aggiornare le statistiche del pianificatore di query.

materialized_view_name

Una vista materializzata. L'istruzione DELETE funziona su una vista materializzata utilizzata per Importazione di dati in streaming. Solo il proprietario della vista materializzata o un utente con privilegio DELETE sulla vista materializzata può eliminare righe dalla vista materializzata.

Non puoi eseguire DELETE su una vista materializzata per l'importazione dati in streaming con una policy di sicurezza a livello di riga (RLS) che non prevede l'autorizzazione IGNORE RLS concessa all'utente. C'è un'eccezione: se all'utente che esegue l'operazione DELETE è stata concessa la licenza IGNORE RLS, l'operazione viene eseguita correttamente. Per ulteriori informazioni, consulta Proprietà e gestione delle policy RLS.

USING table_name, ...

La parola chiave USING viene utilizzata per introdurre un elenco di tabelle quando si fa riferimento a tabelle aggiuntive nella condizione della clausola WHERE. Ad esempio, la seguente istruzione elimina tutte le righe dalla tabella EVENT che soddisfano la condizione di join sulle tabelle EVENT e SALES. La tabella SALES deve essere esplicitamente citata nell'elenco FROM:

```
delete from event using sales where event.eventid=sales.eventid;
```

Se ripeti il nome della tabella di destinazione nella clausola USING, l'operazione DELETE esegue un self join. Puoi utilizzare una sottoquery nella clausola WHERE anziché la sintassi USING come metodo alternativo per scrivere la stessa query.

WHERE condition

Clausola facoltativa che limita l'eliminazione delle righe a quelle che corrispondono alla condizione. Ad esempio, la condizione può essere una restrizione su una colonna, una condizione di join o una condizione basata sul risultato di una query. La query può fare riferimento a tabelle diverse da quella di destinazione del comando DELETE. Ad esempio:

```
delete from t1
where col1 in(select col2 from t2);
```

Se non viene specificata alcuna condizione, vengono eliminate tutte le righe della tabella.

Esempi

Elimina tutte le righe dalla tabella CATEGORY:

```
delete from category;
```

Elimina le righe con valori CATID compresi tra 0 e 9 dalla tabella CATEGORY:

```
delete from category
where catid between 0 and 9;
```

Elimina le righe dalla tabella LISTING i cui valori SELLERID non esistono nella tabella SALES:

```
delete from listing
where listing.sellerid not in(select sales.sellerid from sales);
```

Le due query seguenti eliminano entrambe una riga dalla tabella CATEGORY, in base a un join alla tabella EVENT e a un'ulteriore limitazione sulla colonna CATID:

```
delete from category
using event
where event.catid=category.catid and category.catid=9;
```

```
delete from category
```

```
where catid in
(select category.catid from category, event
where category.catid=event.catid and category.catid=9);
```

La seguente query elimina tutte le righe dalla vista materializzata `mv_cities`. Il nome della vista materializzata in questo esempio è un esempio:

```
delete from mv_cities;
```

DESC DATASHARE

Visualizza un elenco di oggetti di database all'interno di una unità di condivisione dati che vengono aggiunti tramite ALTER DATASHARE. Amazon Redshift visualizza i nomi, i database, gli schemi e i tipi di tabelle, viste e funzioni.

È possibile trovare informazioni aggiuntive sugli oggetti datashare utilizzando le viste di sistema. Per ulteriori informazioni, vedete SVV_DATASHARE_OBJECTS e SVV_DATASHARES.

Sintassi

```
DESC DATASHARE datashare_name [ OF [ ACCOUNT account_id ] NAMESPACE namespace_guid ]
```

Parametri

datashare_name

Il nome dell'unità di condivisione dati.

NAMESPACE namespace_guid

Un valore che specifica lo spazio dei nomi utilizzato dalla unità di condivisione dati. Quando si esegue DESC DATASHARE come amministratore del cluster consumer, è necessario specificare il parametro NAMESPACE per visualizzare le unità di condivisione dati in ingresso.

ACCOUNT account_id

Un valore che specifica l'account a cui appartiene l'unità di condivisione dati.

Note per l'utilizzo

In qualità di amministratore di account consumer, quando esegui DESC DATASHARE per visualizzare le condivisioni di dati in entrata all'interno dell'account, specifica l'opzione NAMESPACE.

AWS Quando esegui DESC DATASHARE per visualizzare le condivisioni di dati in entrata tra gli account, specifica le opzioni ACCOUNT e NAMESPACE. AWS

Esempi

L'esempio seguente mostra le informazioni per le unità di condivisione dati in uscita in un cluster producer.

```
DESC DATASHARE salesshare;

producer_account |          producer_namespace           | share_type  | share_name   |
 object_type |        object_name           |  include_new
-----------------+--------------------------------------+-------------+--------------
+-------------+------------------------------+--------------
 123456789012    | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d  | OUTBOUND    |  salesshare  |
 TABLE       | public.tickit_sales_redshift |
 123456789012    | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d  | OUTBOUND    |  salesshare  |
 SCHEMA      | public                       |   t
```

L'esempio seguente mostra le informazioni per le unità di condivisione dati in entrata in un cluster producer.

```
DESC DATASHARE salesshare of ACCOUNT '123456789012' NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';

 producer_account |          producer_namespace          | share_type | share_name |
 object_type |         object_name          |  include_new
-----------------+--------------------------------------+------------+------------
+-------------+------------------------------+--------------
 123456789012     | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare |
 table       | public.tickit_sales_redshift |
 123456789012     | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d | INBOUND    | salesshare |
 schema      | public                       |
(2 rows)
```

DESC IDENTITY PROVIDER

Visualizza le informazioni relative a un provider di identità. Solo un utente con privilegi avanzati può descrivere un provider di identità.

Sintassi

```
DESC IDENTITY PROVIDER identity_provider_name
```

Parametri

identity_provider_name

Il nome del provider di identità.

Esempio

Nell'esempio seguente vengono visualizzate informazioni sul provider di identità.

```
DESC IDENTITY PROVIDER azure_idp;
```

Output di esempio.

```
  uid   |   name    | type  |             instanceid              | namespc |

                                        params

                          | enabled
--------+-----------+-------+-------------------------------------+---------
+-----------------------------------------------------------------------------------------------------
+---------
 126692 | azure_idp | azure | e40d4bb2-7670-44ae-bfb8-5db013221d73 | aad     |
 {"issuer":"https://login.microsoftonline.com/e40d4bb2-7670-44ae-bfb8-5db013221d73/
v2.0", "client_id":"871c010f-5e61-4fb1-83ac-98610a7e9110", "client_secret":'',
 "audience":["https://analysis.windows.net/powerbi/connector/AmazonRedshift", "https://
analysis.windows.net/powerbi/connector/AWSRDS"]} | t
(1 row)
```

DETACH MASKING POLICY

Scollega da una colonna una policy di mascheramento dinamico dei dati già collegata. Per ulteriori informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.

Una policy di mascheramento può essere scollegata da utenti con privilegi avanzati e da utenti o ruoli che dispongono del ruolo sys:secadmin.

Sintassi

```
DETACH MASKING POLICY policy_name
   ON { table_name }
   ( output_column_names )
   FROM { user_name | ROLE role_name | PUBLIC };
```

Parametri

nome_policy

Nome della policy di mascheramento da scollegare.

table_name

Nome della tabella da cui scollegare policy di mascheramento.

output_column_names

Nomi delle colonne a cui è stata collegata la policy di mascheramento.

user_name

Nome dell'utente a cui è stata collegata la policy di mascheramento.

È possibile impostare solo uno tra user_name, role_name e PUBLIC in una singola istruzione DETACH MASKING POLICY.

role_name

Nome del ruolo a cui è stata collegata la policy di mascheramento.

È possibile impostare solo uno tra user_name, role_name e PUBLIC in una singola istruzione DETACH MASKING POLICY.

PUBLIC

Mostra che la policy è stata collegata a tutti gli utenti nella tabella.

È possibile impostare solo uno tra user_name, role_name e PUBLIC in una singola istruzione DETACH MASKING POLICY.

DETACH RLS POLICY

Scollegamento di una policy di sicurezza a livello di riga su una tabella da uno o più utenti o ruoli.

Una policy può essere scollegata da superuser e utenti o ruoli che dispongono del ruolo `sys:secadmin`.

Sintassi

```
DETACH RLS POLICY policy_name ON [TABLE] table_name [, ...]
FROM { user_name | ROLE role_name | PUBLIC } [, ...]
```

Parametri

nome_policy

Il nome della policy .

ON [TABLE] nome_tabella [, ...]

La tabella o la vista da cui viene scollegata la policy di sicurezza a livello di riga.

FROM { nome_utente | ROLE nome_ruolo | PUBLIC} [, ...]

Specifica se la policy è scollegata da uno o più utenti o ruoli specificati.

Note per l'utilizzo

Quando lavori con l'istruzione DETACH RLS POLICY, tieni presente quanto segue:

- Puoi scollegare una policy da una relazione, un utente, un ruolo o un pubblico.

Esempi

Nell'esempio seguente, una policy su una tabella viene scollegata da un ruolo.

```
DETACH RLS POLICY policy_concerts ON tickit_category_redshift FROM ROLE analyst, ROLE
 dbadmin;
```

DROP DATABASE

Rimuove un database.

Non è possibile eseguire DROP DATABASE all'interno di un blocco di transazione (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Sintassi

```
DROP DATABASE database_name
```

Parametri

database_name

Nome del database da rimuovere Non puoi rilasciare i database dev, padb_harvest, template0, template1 o sys:internal e non puoi rilasciare il database corrente.

Per rimuovere un database esterno, elimina lo schema esterno. Per ulteriori informazioni, consulta DROP SCHEMA.

Note sull'utilizzo di DROP DATABASE

Durante l'utilizzo dell'istruzione DROP DATABASE, considera quando segue:

- In generale, si consiglia di non eliminare un database che contiene un datashare utilizzando l'istruzione DROP DATABASE. AWS Data Exchange Se lo fai, chi Account AWS ha accesso al datashare perde l'accesso. L'esecuzione di questo tipo di alterazione può violare i termini del prodotto dei dati in AWS Data Exchange.

  L'esempio seguente mostra un errore quando un database che contiene un AWS Data Exchange datashare viene eliminato.

  ```
  DROP DATABASE test_db;
  ERROR:   Drop of database test_db that contains ADX-managed datashare(s)
   requires session variable datashare_break_glass_session_var to be set to value
   'ce8d280c10ad41'
  ```

  Per consentire l'eliminazione del database, impostare la variabile seguente ed eseguire nuovamente l'istruzione DROP DATABASE.

  ```
  SET datashare_break_glass_session_var to 'ce8d280c10ad41';
  ```

  ```
  DROP DATABASE test_db;
  ```

In questo caso, Amazon Redshift genera un valore casuale una tantum per impostare la variabile di sessione per consentire DROP DATASHARE per un database che contiene un'unità di condivisione dati AWS Data Exchange .

Esempi

L'esempio seguente rimuove un database denominato TICKIT_TEST:

```
drop database tickit_test;
```

DROP DATASHARE

Elimina una unità di condivisione dati. Questo comando è irreversibile.

Solo un utente con privilegi avanzati o il proprietario di una unità di condivisione dati può eliminare una unità di condivisione dati.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per DROP DATASHARE:

- Superuser
- Utenti con il privilegio DROP DATASHARE
- Proprietario dell'unità di condivisione dati

Sintassi

```
DROP DATASHARE datashare_name;
```

Parametri

datashare_name

Il nome dell'unità di condivisione dati da rimuovere.

Note per l'utilizzo di DROP DATASHARE

Durante l'utilizzo dell'istruzione DROP DATABASE, considera quando segue:

- In generale, si consiglia di non eliminare un AWS Data Exchange datashare utilizzando l'istruzione DROP DATASHARE. Se lo fai, chi ha accesso al Account AWS datashare perde l'accesso. L'esecuzione di questo tipo di alterazione può violare i termini del prodotto dei dati in AWS Data Exchange.

  L'esempio seguente mostra un errore quando un AWS Data Exchange datashare viene eliminato.

  ```
  DROP DATASHARE salesshare;
  ERROR:  Drop of ADX-managed datashare salesshare requires session variable
   datashare_break_glass_session_var to be set to value '620c871f890c49'
  ```

  Per consentire l'eliminazione di un AWS Data Exchange datashare, impostate la variabile seguente ed eseguite nuovamente l'istruzione DROP DATASHARE.

  ```
  SET datashare_break_glass_session_var to '620c871f890c49';
  ```

  ```
  DROP DATASHARE salesshare;
  ```

  In questo caso, Amazon Redshift genera un valore casuale una tantum per impostare la variabile di sessione in modo da consentire DROP DATASHARE per un datashare. AWS Data Exchange

Esempi

Nell'esempio seguente viene rimossa un'unità di condivisione dati denominata `salesshare`.

```
DROP DATASHARE salesshare;
```

DROP EXTERNAL VIEW (anteprima)

Questa è una documentazione di pre-rilascio per le viste del Catalogo dati per Amazon Redshift, che è nella versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche . Consigliamo di utilizzare questa funzione solo con cluster di test e non in ambienti di produzione. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms.

È possibile creare un cluster Amazon Redshift di anteprima per testare nuove funzionalità di Amazon Redshift. Non è possibile utilizzare queste funzionalità in produzione o spostare il cluster di

anteprima in un cluster di produzione o in un cluster su un'altra traccia. Per i termini e le condizioni dell'anteprima, consulta la sezione relativa a beta e anteprime nei termini del servizio AWS.

Come creare un cluster di anteprima

1. Accedi AWS Management Console e apri la console Amazon Redshift all'indirizzo https://console.aws.amazon.com/redshiftv2/.
2. Dal menu di navigazione, scegli Provisioned clusters dashboard (Pannello di controllo dei cluster con provisioning) e seleziona Clusters (Cluster). Regione AWS Sono elencati i cluster del tuo account nella versione corrente. Nelle colonne dell'elenco è visualizzato un sottoinsieme delle proprietà di ciascun cluster.
3. Nella pagina dell'elenco dei cluster viene visualizzato un banner che introduce l'anteprima. Scegli il pulsante Create preview cluster (Crea cluster di anteprima) per aprire la pagina di creazione del cluster.
4. Inserisci le proprietà del cluster. Scegli la traccia di anteprima che contiene le funzionalità che desideri testare. Consigliamo di assegnare al cluster un nome che indichi che si trova in una traccia di anteprima. Scegli le opzioni per il cluster, tra cui quelle contrassegnate come - anteprima, per le funzionalità che desideri testare. Per informazioni generali sulla creazione di cluster, consulta Creazione di un cluster nella Guida alla gestione di Amazon Redshift.
5. Per creare un cluster di anteprima, scegli Crea cluster.

   > **Note**
   >
   > La traccia `preview_2023` è la traccia di anteprima più recente disponibile. Questa traccia supporta la creazione di cluster solo con tipi di nodo RA3. Il tipo di nodo DC2 e qualsiasi tipo di nodo precedente non sono supportati.

6. Quando il cluster di anteprima è disponibile, utilizza il client SQL per caricare dati ed eseguire query su di essi.

La funzionalità di anteprima delle viste del catalogo dati è disponibile solo nelle seguenti regioni.

- Stati Uniti orientali (Ohio) (us-east-2)
- Stati Uniti orientali (Virginia settentrionale) (us-east-1)
- Stati Uniti occidentali (California settentrionale) (us-west-1)
- Asia Pacifico (Tokyo) (ap-northeast-1)

- Europa (Irlanda) (eu-west-1)
- Europa (Stoccolma) (eu-north-1)

Puoi anche creare un gruppo di lavoro di anteprima per testare le viste del catalogo dati. Non è possibile utilizzare queste funzionalità in produzione o trasferire il gruppo di lavoro in un altro gruppo di lavoro. Per i termini e le condizioni dell'anteprima, consulta Beta and Previews in AWS Service Terms. Per istruzioni su come creare un gruppo di lavoro di anteprima, consulta https://docs.aws.amazon.com/redshift/latest/mgmt/serverless-workgroup-preview.html.

Rilascia una vista esterna dal database. Il rilascio di una vista esterna la rimuove da tutti i motori SQL, come Amazon Athena e Amazon EMR Spark, a cui è associata. Questo comando non può essere annullato. Per ulteriori informazioni sulle viste del catalogo dati, consulta Creating Data Catalog views (preview).

Sintassi

```
DROP EXTERNAL VIEW schema_name.view_name [ IF EXISTS ]
{catalog_name.schema_name.view_name | awsdatacatalog.dbname.view_name |
 external_schema_name.view_name}
```

Parametri

schema_name.view_name

Lo schema allegato al AWS Glue database, seguito dal nome della vista.

IF EXISTS

Rilascia la vista solo se esiste.

catalog_name.schema_name.view_name | awsdatacatalog.dbname.view_name | external_schema_name.view_name

La notazione dello schema da usare per il rilascio della vista. È possibile specificare di utilizzare il AWS Glue Data Catalog, un database Glue creato dall'utente o uno schema esterno creato dall'utente. Per ulteriori informazioni, consulta CREATE DATABASE e CREATE EXTERNAL SCHEMA.

query_definition

La definizione della query SQL che Amazon Redshift esegue per alterare la vista.

Esempi

L'esempio seguente rilascia una vista del catalogo dati denominata sample_schema.glue_data_catalog_view.

```
DROP EXTERNAL VIEW sample_schema.glue_data_catalog_view IF EXISTS
```

DROP FUNCTION

Rimuove una funzione definita dall'utente (UDF, user-defined function) dal database. La firma della funzione o l'elenco dei tipi di dati dell'argomento devono essere specificati perché possono esistere più funzioni con lo stesso nome ma diverse firme. Non è possibile rimuovere una funzione integrata di Amazon Redshift.

Questo comando è irreversibile.

Privilegi richiesti

Di seguito sono elencati i privilegi richiesti per DROP FUNCTION:

- Superuser
- Utenti con il privilegio DROP FUNCTION
- Proprietario della funzione

Sintassi

```
DROP FUNCTION name
( [arg_name] arg_type   [, ...] )
[ CASCADE | RESTRICT ]
```

Parametri

name

Il nome della funzione da rimuovere.

arg_name

Il nome di un argomento di input. DROP FUNCTION ignora i nomi degli argomenti perché sono necessari solo i tipi di dati dell'argomento per determinare l'identità della funzione.

arg_type

Tipo di dati dell'argomento di input. Puoi fornire un elenco separato da virgole con un massimo di 32 tipi di dati.

CASCADE

Parola chiave che specifica di rimuovere automaticamente gli oggetti che dipendono dalla funzione, come le viste.

Per creare una vista che non dipende da una funzione, includi la clausola WITH NO SCHEMA BINDING nella definizione della vista. Per ulteriori informazioni, consulta CREATE VIEW.

RESTRICT

Parola chiave che specifica che se un oggetto dipende dalla funzione, non rimuove la funzione e restituisce un messaggio. Questa operazione costituisce l'impostazione predefinita.

Esempi

L'esempio seguente rimuove la funzione denominata `f_sqrt`:

```
drop function f_sqrt(int);
```

Per rimuovere una funzione con dipendenze, utilizza l'opzione CASCADE, come mostrato nell'esempio seguente:

```
drop function f_sqrt(int)cascade;
```

DROP GROUP

Elimina un gruppo di utenti. Questo comando è irreversibile. Questo comando non elimina i singoli utenti in un gruppo.

Vedi DROP USER per eliminare un singolo utente.

Sintassi

```
DROP GROUP name
```

Parametro

name

Nome del gruppo di utenti da eliminare.

Esempio

L'esempio seguente elimina il gruppo di `guests` utenti:

```
DROP GROUP guests;
```

Non puoi rimuovere un gruppo se il gruppo ha privilegi su un oggetto. Se tenti di rimuovere un gruppo di questo tipo, verrà restituito il seguente errore.

```
ERROR: group "guests" can't be dropped because the group has a privilege on some object
```

Se il gruppo dispone di privilegi per un oggetto, è necessario revocare i privilegi prima di eliminare il gruppo. Per trovare gli oggetti per i quali il `guests` gruppo dispone dei privilegi, utilizzate l'esempio seguente. Per ulteriori informazioni sulla visualizzazione dei metadati utilizzata nell'esempio, vedete SVV_RELATION_PRIVILEGES.

```
SELECT DISTINCT namespace_name, relation_name, identity_name, identity_type
FROM svv_relation_privileges
WHERE identity_type='group' AND identity_name='guests';

+----------------+---------------+---------------+---------------+
| namespace_name | relation_name | identity_name | identity_type |
+----------------+---------------+---------------+---------------+
| public         | table1        | guests        | group         |
+----------------+---------------+---------------+---------------+
| public         | table2        | guests        | group         |
+----------------+---------------+---------------+---------------+
```

L'esempio seguente revoca tutti i privilegi su tutte le tabelle nello schema `public` dal gruppo di utenti `guests` e quindi rimuove il gruppo.

```
REVOKE ALL ON ALL TABLES IN SCHEMA public FROM GROUP guests;
DROP GROUP guests;
```

DROP IDENTITY PROVIDER

Elimina un provider di identità. Questo comando è irreversibile. Solo un utente con privilegi avanzati può eliminare un provider di identità.

Sintassi

```
DROP IDENTITY PROVIDER identity_provider_name [ CASCADE ]
```

Parametri

identity_provider_name

Il nome del provider di identità da eliminare.

CASCADE

Elimina utenti e ruoli collegati al provider di identità, quando questo viene eliminato.

Esempio

L'esempio seguente elimina il provider di identità oauth_provider.

```
DROP IDENTITY PROVIDER oauth_provider;
```

Se si elimina il provider di identità, alcuni utenti potrebbero non essere in grado di accedere o utilizzare gli strumenti client configurati per utilizzare il provider di identità.

DROP LIBRARY

Rimuove una libreria Python personalizzata dal database. Solo il proprietario della libreria o un utente con privilegi avanzati può rimuovere una libreria.

DROP LIBRARY non può essere eseguito all'interno di un blocco di transazioni (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Questo comando è irreversibile. Il comando DROP LIBRARY viene immediatamente sottoposto al commit. Se una UDF che dipende dalla libreria viene eseguita contemporaneamente, l'UDF potrebbe non riuscire anche se è in esecuzione all'interno di una transazione.

Per ulteriori informazioni, consulta CREATE LIBRARY.

Privilegi richiesti

Di seguito sono elencati i privilegi richiesti per DROPLIBRARY:

- Superuser
- Utenti con il privilegio DROP LIBRARY
- Proprietario della libreria

Sintassi

```
DROP LIBRARY library_name
```

Parametri

library_name

Nome della libreria.

DROP MASKING POLICY

Elimina una policy di mascheramento dinamico dei dati da tutti i database. Non è possibile eliminare una policy di mascheramento ancora collegata a una o più tabelle. Per ulteriori informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.

Una policy di mascheramento può essere eliminata da utenti con privilegi avanzati e da utenti o ruoli che dispongono del ruolo sys:secadmin.

Sintassi

```
DROP MASKING POLICY policy_name;
```

Parametri

nome_policy

Nome della policy di mascheramento da eliminare.

DROP MODEL

Rimuove un modello dal database. Solo il proprietario del modello o un utente con privilegi avanzati può rimuovere un modello.

DROP MODEL elimina anche tutte le funzioni di previsione associate derivate da questo modello, tutti gli artefatti di Amazon Redshift correlati al modello e tutti i dati Amazon S3 relativi al modello. Mentre il modello è ancora in fase di addestramento in Amazon SageMaker, DROP MODEL annullerà tali operazioni.

Questo comando è irreversibile. Il comando DROP MODEL esegue immediatamente il commit.

Autorizzazioni richieste

Di seguito sono riportate le autorizzazioni richieste per DROP MODEL:

- Superuser
- Utenti con autorizzazione DROP MODEL
- Proprietario del modello
- Proprietario dello schema

Sintassi

```
DROP MODEL [ IF EXISTS ] model_name
```

Parametri

IF EXISTS

Una clausola che indica che se lo schema specificato esiste già, il comando non deve apportare modifiche e deve restituire un messaggio che lo schema esiste.

model_name

Il nome del modello. Il nome del modello in uno schema deve essere unico.

Esempi

L'esempio seguente rimuove il modello demo_ml.customer_churn.

```
DROP MODEL demo_ml.customer_churn
```

DROP MATERIALIZED VIEW

Elimina una vista materializzata

Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

Sintassi

```
DROP MATERIALIZED VIEW [ IF EXISTS ] mv_name [ CASCADE | RESTRICT ]
```

Parametri

IF EXISTS

Una clausola che specifica di verificare se esiste la vista materializzata denominata. Se la vista materializzata non esiste, il comando `DROP MATERIALIZED VIEW` restituisce un messaggio di errore. Questa clausola è utile all'interno degli script, per evitare che lo script fallisca in caso di eliminazione di una vista materializzata inesistente.

mv_name

Il nome della vista materializzata da eliminare.

CASCADE

Una clausola che indica di eliminare automaticamente gli oggetti da cui dipende la vista materializzata, come altre viste.

RESTRICT

Una clausola che indica di non eliminare la vista materializzata se alcuni oggetti dipendono da essa. Questa è l'impostazione predefinita.

Note per l'utilizzo

Solo il proprietario di una vista materializzata può utilizzare `DROP MATERIALIZED VIEW` su quella vista. Un utente con privilegi avanzati o un utente a cui sono stati specificamente concessi i privilegi DROP possono costituire un'eccezione.

Quando scrivi un'istruzione drop per una vista materializzata ed esiste una vista con un nome corrispondente, viene generato un errore che indica di utilizzare DROP VIEW. Si verifica un errore anche nel caso in cui si utilizza `DROP MATERIALIZED VIEW IF EXISTS`.

Esempio

Nell'esempio seguente viene eseguita l'eliminazione della vista materializzata `tickets_mv`.

```
DROP MATERIALIZED VIEW tickets_mv;
```

DROP PROCEDURE

Rilascia una procedura. Per rilasciare una procedura, sono necessari sia il nome della procedura che i tipi di dati dell'argomento di input (firma). Facoltativamente, puoi includere i tipi di dati dell'argomento completi, inclusi gli argomenti OUT. Per trovare la firma di una procedura, utilizza il comando SHOW PROCEDURE. Per ulteriori informazioni sulle firme delle procedure, consulta PG_PROC_INFO.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per DROP PROCEDURE:

- Superuser
- Utenti con il privilegio DROP PROCEDURE
- Proprietario della procedura

Sintassi

```
DROP PROCEDURE sp_name ( [ [ argname ] [ argmode ] argtype [, ...] ] )
```

Parametri

sp_name

Il nome della procedura da rimuovere.

argname

Il nome di un argomento di input. DROP PROCEDURE ignora i nomi degli argomenti, poiché sono necessari solo i tipi di dati dell'argomento per determinare l'identità della funzione.

argmode

La modalità di un argomento, che può essere IN, OUT o INOUT. Gli argomenti OUT sono facoltativi in quanto non vengono utilizzati per individuare una procedura archiviata.

argtype

Tipo di dati dell'argomento di input. Per un elenco dei tipi di dati supportati, consultare Tipi di dati.

Esempi

L'esempio seguente rilascia una procedura archiviata denominata `quarterly_revenue`.

```
DROP PROCEDURE quarterly_revenue(volume INOUT bigint, at_price IN numeric,result OUT
 int);
```

DROP RLS POLICY

Elimina una policy di sicurezza a livello di riga per tutte le tabelle in tutti i database.

Una policy può essere eliminata da superuser e utenti o ruoli che dispongono del ruolo sys:secadmin.

Sintassi

```
DROP RLS POLICY [ IF EXISTS ] policy_name [ CASCADE | RESTRICT ]
```

Parametri

IF EXISTS

Una clausola che indica se la policy specificata esiste già.

nome_policy

Il nome della policy .

CASCADE

Una clausola che indica di scollegare automaticamente la policy da tutte le tabelle collegate prima di eliminarla.

RESTRICT

Una clausola che indica di non eliminare la policy quando viene collegata ad alcune tabelle. Questa è l'impostazione predefinita.

Esempi

Nell'esempio seguente viene eliminata la policy di sicurezza a livello di riga.

```
DROP RLS POLICY policy_concerts;
```

DROP ROLE

Rimuove un ruolo da un database. Solo il proprietario del ruolo che ha creato il ruolo, un utente con l'opzione WITH ADMIN o un utente con privilegi avanzati può eliminare un ruolo.

Non è possibile eliminare un ruolo concesso a un utente o un altro ruolo dipendente da questo ruolo.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per DROP ROLE:

- Superuser
- Proprietario del ruolo che è l'utente che ha creato il ruolo o un utente a cui è stato concesso il ruolo con il privilegio WITH ADMIN OPTION.

Sintassi

```
DROP ROLE role_name [ FORCE | RESTRICT ]
```

Parametri

role_name

Il nome del ruolo.

[FORCE | RESTRICT]

L'impostazione predefinita è RESTRICT. Amazon Redshift genera un errore quando si tenta di eliminare un ruolo che ha ereditato un altro ruolo. Usa FORCE per rimuovere tutte le assegnazioni del ruolo, se esistenti.

Esempi

L'esempio seguente rimuove il ruolo `sample_role`.

```
DROP ROLE sample_role FORCE;
```

L'esempio seguente tenta di eliminare il ruolo sample_role1 concesso a un utente con l'opzione RESTRICT di default.

```
CREATE ROLE sample_role1;
GRANT sample_role1 TO user1;
DROP ROLE sample_role1;
ERROR:  cannot drop this role since it has been granted on a user
```

Per eliminare correttamente il sample_role1 concesso a un utente, utilizza l'opzione FORCE.

```
DROP ROLE sample_role1 FORCE;
```

L'esempio seguente tenta di eliminare il ruolo sample_role2 che ha un altro ruolo dipendente da esso con l'opzione RESTRICT di default.

```
CREATE ROLE sample_role1;
CREATE ROLE sample_role2;
GRANT sample_role1 TO sample_role2;
DROP ROLE sample_role2;
ERROR:  cannot drop this role since it depends on another role
```

Per eliminare correttamente il sample_role2, da cui dipende un altro ruolo, utilizza l'opzione FORCE.

```
DROP ROLE sample_role2 FORCE;
```

DROP SCHEMA

Elimina uno schema. Per uno schema esterno, puoi anche rimuovere il database esterno associato allo schema. Questo comando è irreversibile.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per DROP SCHEMA:

- Superuser
- Proprietario dello schema
- Utenti con il privilegio DROP SCHEMA

Sintassi

```
DROP SCHEMA [ IF EXISTS ] name [, ...]
[ DROP EXTERNAL DATABASE ]
[ CASCADE | RESTRICT ]
```

Parametri

IF EXISTS

Clausola che indica che se lo schema specificato non esiste, il comando non deve apportare modifiche e deve restituire un messaggio che lo schema non esiste, piuttosto che terminare con un errore.

Questa clausola è utile durante lo scripting, quindi lo script ha esito positivo se DROP SCHEMA viene eseguito su uno schema inesistente.

name

Nomi degli schemi da rimuovere. Puoi specificare diversi nomi di schema separati da virgole.

DROP EXTERNAL DATABASE

Una clausola che indica che se uno schema esterno è stato rimosso, devi rimuovere il database esterno associato allo schema esterno, se esiste. Se non esiste alcun database esterno, il comando restituisce un messaggio che dichiara che non esiste alcun database esterno. Se

vengono rimossi più schemi esterni, tutti i database associati agli schemi specifici vengono eliminati.

Se un database esterno contiene oggetti dipendenti come le tabelle, includi l'opzione CASCADE per rimuovere gli oggetti dipendenti.

Quando rimuovi un database esterno, il database viene rimosso anche per altri schemi associati al database. Vengono anche rimosse le tabelle definite in altri schemi esterni tramite il database.

DROP EXTERNAL DATABASE non supporta i database esterni archiviati in un metastore HIVE.

CASCADE

Parola chiave che indica di rimuovere automaticamente tutti gli oggetti nello schema. Se viene specificato DROP EXTERNAL DATABASE, anche tutti gli oggetti nel database esterno vengono rimossi.

RESTRICT

Parola chiave che indica di non eliminare uno schema o database esterno se contiene oggetti. Questa operazione costituisce l'impostazione predefinita.

Esempio

L'esempio seguente elimina uno schema denominato S_SALES. Questo esempio utilizza RESTRICT come meccanismo di sicurezza in modo che lo schema non venga eliminato se contiene oggetti. In questo caso, è necessario eliminare gli oggetti dello schema prima di eliminare lo schema.

```
drop schema s_sales restrict;
```

L'esempio seguente elimina uno schema denominato S_SALES e tutti gli oggetti che dipendono da tale schema.

```
drop schema s_sales cascade;
```

L'esempio seguente rimuove lo schema S_SALES se esiste oppure non esegue nulla e restituisce un messaggio se non esiste.

```
drop schema if exists s_sales;
```

L'esempio seguente elimina uno schema esterno denominato S_SPECTRUM e il database esterno a esso associato. Questo esempio utilizza RESTRICT in modo che lo schema e il database non vengano eliminati se contengono oggetti. In questo caso, è necessario eliminare gli oggetti dipendenti prima di eliminare lo schema e il database.

```
drop schema s_spectrum drop external database restrict;
```

L'esempio seguente elimina più schemi e i database esterni a essi associati, insieme agli oggetti dipendenti.

```
drop schema s_sales, s_profit, s_revenue drop external database cascade;
```

DROP TABLE

Rimuove una tabella da un database.

Se stai tentando di svuotare una tabella di righe, senza rimuovere la tabella, utilizza il comando DELETE o TRUNCATE.

DROP TABLE rimuove i vincoli esistenti sulla tabella di destinazione. È possibile rimuovere più tabelle con un singolo comando DROP TABLE.

Non è possibile eseguire DROP TABLE con una tabella esterna all'interno di una transazione (BEGIN … END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Per trovare un esempio in cui il privilegio DROP è concesso a un gruppo, consultare GRANT Esempi.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per DROP TABLE:

- Superuser
- Utenti con il privilegio DROP TABLE
- Il proprietario della tabella con il privilegio USAGE sullo schema

Sintassi

```
DROP TABLE [ IF EXISTS ] name [, ...] [ CASCADE | RESTRICT ]
```

Parametri

IF EXISTS

Clausola che indica che se la tabella specificata non esiste, il comando non deve apportare modifiche e deve restituire un messaggio che la tabella non esiste, piuttosto che terminare con un errore.

Questa clausola è utile durante lo scripting, quindi lo script ha esito positivo se DROP TABLE viene eseguito su una tabella inesistente.

name

Nome della tabella da rimuovere.

CASCADE

Clausola che indica di rimuovere automaticamente gli oggetti che dipendono dalla tabella, come le viste.

Per creare una vista che non dipende da altri oggetti del database, come viste e tabelle, includi la clausola WITH NO SCHEMA BINDING nella definizione della vista. Per ulteriori informazioni, consulta CREATE VIEW.

RESTRICT

Clausola che indica di non eliminare la tabella se sono presenti oggetti che dipendono da essa. Questa operazione costituisce l'impostazione predefinita.

Esempi

Rimozione di una tabella senza dipendenze

L'esempio seguente crea e rimuove una tabella denominata FEEDBACK che non ha dipendenze:

```
create table feedback(a int);

drop table feedback;
```

Se una tabella contiene colonne a cui fanno riferimento le viste o altre tabelle, Amazon Redshift visualizza un messaggio come il seguente.

```
Invalid operation: cannot drop table feedback because other objects depend on it
```

Rimozione simultanea di due tabelle

Il seguente set di comandi crea una tabella FEEDBACK e una tabella BUYERS e quindi elimina entrambe le tabelle con un singolo comando:

```
create table feedback(a int);

create table buyers(a int);

drop table feedback, buyers;
```

Rimozione di una tabella con una dipendenza

Le seguenti operazioni mostrano come rimuovere una tabella denominata FEEDBACK utilizzando l'opzione CASCADE.

Innanzitutto, crea una semplice tabella denominata FEEDBACK utilizzando il comando CREATE TABLE:

```
create table feedback(a int);
```

Quindi, utilizza il comando CREATE VIEW per creare una vista denominata FEEDBACK_VIEW che si basa sulla tabella FEEDBACK:

```
create view feedback_view as select * from feedback;
```

L'esempio seguente rimuove la tabella FEEDBACK e rimuove anche la vista FEEDBACK_VIEW, perché FEEDBACK_VIEW dipende dalla tabella FEEDBACK:

```
drop table feedback cascade;
```

Visualizzazione delle dipendenze per una tabella

Per restituire le dipendenze per la tabella, utilizza l'esempio seguente. Sostituisci *my_schema* e *my_table* con il tuo schema e la tua tabella.

```
SELECT dependent_ns.nspname as dependent_schema
, dependent_view.relname as dependent_view
, source_ns.nspname as source_schema
, source_table.relname as source_table
, pg_attribute.attname as column_name
```

```
FROM pg_depend
JOIN pg_rewrite ON pg_depend.objid = pg_rewrite.oid
JOIN pg_class as dependent_view ON pg_rewrite.ev_class = dependent_view.oid
JOIN pg_class as source_table ON pg_depend.refobjid = source_table.oid
JOIN pg_attribute ON pg_depend.refobjid = pg_attribute.attrelid
    AND pg_depend.refobjsubid = pg_attribute.attnum
JOIN pg_namespace dependent_ns ON dependent_ns.oid = dependent_view.relnamespace
JOIN pg_namespace source_ns ON source_ns.oid = source_table.relnamespace
WHERE
source_ns.nspname = 'my_schema'
AND source_table.relname = 'my_table'
AND pg_attribute.attnum > 0
ORDER BY 1,2
LIMIT 10;
```

Per rilasciare *my_table* e le sue dipendenze, usa il seguente esempio. Questo esempio restituisce anche tutte le dipendenze per la tabella che è stata rilasciata.

```
DROP TABLE my_table CASCADE;

SELECT dependent_ns.nspname as dependent_schema
, dependent_view.relname as dependent_view
, source_ns.nspname as source_schema
, source_table.relname as source_table
, pg_attribute.attname as column_name
FROM pg_depend
JOIN pg_rewrite ON pg_depend.objid = pg_rewrite.oid
JOIN pg_class as dependent_view ON pg_rewrite.ev_class = dependent_view.oid
JOIN pg_class as source_table ON pg_depend.refobjid = source_table.oid
JOIN pg_attribute ON pg_depend.refobjid = pg_attribute.attrelid
    AND pg_depend.refobjsubid = pg_attribute.attnum
JOIN pg_namespace dependent_ns ON dependent_ns.oid = dependent_view.relnamespace
JOIN pg_namespace source_ns ON source_ns.oid = source_table.relnamespace
WHERE
source_ns.nspname = 'my_schema'
AND source_table.relname = 'my_table'
AND pg_attribute.attnum > 0
ORDER BY 1,2
LIMIT 10;

+------------------+----------------+---------------+--------------+-------------+
| dependent_schema | dependent_view | source_schema | source_table | column_name |
+------------------+----------------+---------------+--------------+-------------+
```

Rimozione di una tabella utilizzando IF EXISTS

L'esempio seguente rimuove la tabella FEEDBACK se esiste oppure non esegue nulla e restituisce un messaggio se non esiste:

```
drop table if exists feedback;
```

DROP USER

Rimuove un utente da un database. È possibile rimuovere più utenti con un singolo comando DROP USER. È necessario essere un superutente del database o disporre dell'autorizzazione DROP USER per eseguire questo comando.

Sintassi

```
DROP USER [ IF EXISTS ] name [, ... ]
```

Parametri

IF EXISTS

Clausola che stabilisce che se l'utente specificato non esiste, il comando non deve apportare modifiche e deve restituire un messaggio indicante che l'utente non esiste, piuttosto che terminare con un errore.

Questa clausola è utile durante lo scripting, in modo che lo script abbia esito positivo se DROP USER viene eseguito su un utente inesistente.

name

Nome dell'utente da rimuovere. Puoi specificare più utenti, usando una virgola per separare ciascun nome dal successivo.

Note per l'utilizzo

Non puoi eliminare l'utente denominato `rdsdb` o l'utente amministratore del database che in genere è denominato `awsuser` o `admin`.

Non puoi rimuovere un utente se questo possiede un oggetto di database, come uno schema, un database, una tabella o una vista oppure se l'utente ha privilegi su un database, una tabella, una

colonna o un gruppo. Se tenti di rimuovere un utente di questo tipo, verrà restituito uno dei seguenti errori.

```
ERROR: user "username" can't be dropped because the user owns some object [SQL
 State=55006]

ERROR: user "username" can't be dropped because the user has a privilege on some object
 [SQL State=55006]
```

Per istruzioni dettagliate su come trovare gli oggetti di proprietà di un utente del database, consulta How do I resolve the "user cannot be dropped" error in Amazon Redshift? (Come risolvere l'errore "l'utente non può essere eliminato" in Amazon Redshift?) nel Knowledge Center.

> **Note**
>
> Amazon Redshift controlla solo il database corrente prima di rimuovere un utente. DROP USER non restituisce un errore se l'utente possiede oggetti di database o ha privilegi su oggetti in un altro database. Se rimuovi un utente proprietario di oggetti in un altro database, il proprietario di tali oggetti viene modificato in 'unknown'.

Se un utente è proprietario di un oggetto, prima rimuovi l'oggetto o passa la proprietà a un altro utente prima di rimuovere l'utente originale. Se l'utente ha i privilegi per un oggetto, revoca i privilegi prima di rimuovere l'utente. L'esempio seguente mostra come rimuovere un oggetto, modificare la proprietà e revocare i privilegi prima di rimuovere l'utente.

```
drop database dwdatabase;
alter schema dw owner to dwadmin;
revoke all on table dwtable from dwuser;
drop user dwuser;
```

Esempi

L'esempio seguente rimuove un utente chiamato paulo:

```
drop user paulo;
```

L'esempio seguente rimuove due utenti, paulo e martha:

```
drop user paulo, martha;
```

L'esempio seguente rimuove l'utente paulo, se esiste, oppure non viene eseguita alcuna operazione e viene restituito un messaggio se l'utente è inesistente:

```
drop user if exists paulo;
```

DROP VIEW

Rimuove una vista dal database. È possibile rimuovere più viste con un singolo comando DROP VIEW. Questo comando è irreversibile.

Privilegi richiesti

Di seguito sono riportati i privilegi richiesti per DROP VIEW:

- Superuser
- Utenti con il privilegio DROP VIEW
- Proprietario della visualizzazione

Sintassi

```
DROP VIEW [ IF EXISTS ] name [, ... ] [ CASCADE | RESTRICT ]
```

Parametri

IF EXISTS

Clausola che indica che se la vista specificata non esiste, il comando non deve apportare modifiche e deve restituire un messaggio che la vista non esiste, piuttosto che terminare con un errore.

Questa clausola è utile durante lo scripting, quindi lo script ha esito positivo se DROP VIEW viene eseguito su una vista inesistente.

name

Nome della vista da rimuovere.

CASCADE

Clausola che indica di rimuovere automaticamente gli oggetti che dipendono dalla vista, come altre viste.

Per creare una vista che non dipende da altri oggetti del database, come viste e tabelle, includi la clausola WITH NO SCHEMA BINDING nella definizione della vista. Per ulteriori informazioni, consulta CREATE VIEW.

Tieni presente che se includi CASCADE e il numero di oggetti del database eliminati è pari o superiore a dieci, è possibile che il client del database non elenchi tutti gli oggetti eliminati nei risultati di riepilogo. Ciò è in genere dovuto al fatto che gli strumenti client SQL hanno limitazioni predefinite sui risultati restituiti.

RESTRICT

Clausola che indica di non eliminare la vista se sono presenti oggetti che dipendono da essa. Questa operazione costituisce l'impostazione predefinita.

Esempi

L'esempio seguente rimuove la vista denominata event:

```
drop view event;
```

Per rimuovere una vista con dipendenze, utilizza l'opzione CASCADE. Ad esempio, supponiamo di iniziare con una tabella denominata EVENT. Creiamo quindi la vista eventview della tabella EVENT, usando il comando CREATE VIEW, come mostrato nel seguente esempio:

```
create view eventview as
select dateid, eventname, catid
from event where catid = 1;
```

Ora creiamo una seconda vista chiamata myeventview, che si basa sulla prima vista eventview:

```
create view myeventview as
select eventname, catid
from eventview where eventname <> ' ';
```

A questo punto, sono state create due viste: eventview e myeventview.

La vista myeventview è una vista figlio con eventview come padre.

Per eliminare la vista eventview, il comando da usare è il seguente:

```
drop view eventview;
```

Tieni presente che se esegui il comando in questo caso, viene restituito il seguente errore:

```
drop view eventview;
ERROR: can't drop view eventview because other objects depend on it
HINT: Use DROP ... CASCADE to drop the dependent objects too.
```

Per rimediare, emettere il seguente comando (come suggerito nel messaggio di errore):

```
drop view eventview cascade;
```

A questo punto la rimozione delle due viste eventview e myeventview è stata completata.

Nell'esempio seguente viene rimossa la vista eventview se esiste oppure non viene eseguita alcuna operazione e viene restituito un messaggio se non esiste:

```
drop view if exists eventview;
```

END

Esegue il commit della transazione corrente. Esegue esattamente la stessa funzione del comando COMMIT.

Per una documentazione più dettagliata, consultare COMMIT.

Sintassi

```
END [ WORK | TRANSACTION ]
```

Parametri

WORK

Parola chiave facoltativa.

TRANSACTION

Parola chiave facoltativa; WORK e TRANSACTION sono sinonimi.

Esempi

Gli esempi seguenti terminano tutti il blocco della transazione ed eseguono il commit della transazione:

```
end;
```

```
end work;
```

```
end transaction;
```

Dopo uno di questi comandi, Amazon Redshift termina il blocco della transazione ed esegue il commit delle modifiche.

EXECUTE

Esegue un'istruzione preparata in precedenza.

Sintassi

```
EXECUTE plan_name [ (parameter [, ...]) ]
```

Parametri

plan_name

Il nome dell'istruzione da preparare.

parameter

Il valore effettivo di un parametro dell'istruzione preparata. Deve essere un'espressione che restituisce un valore di un tipo compatibile con il tipo di dati specificato per questa posizione del parametro nel comando PREPARE che ha creato l'istruzione preparata.

Note per l'utilizzo

EXECUTE viene utilizzato per eseguire un'istruzione preparata in precedenza. Poiché le istruzioni preparate esistono solo per la durata di una sessione, l'istruzione preparata deve essere stata creata da un'istruzione PREPARE eseguita in precedenza nella sessione corrente.

Se la precedente istruzione PREPARE ha specificato alcuni parametri, un set di parametri compatibile deve essere passato all'istruzione EXECUTE, altrimenti Amazon Redshift restituisce un errore. A differenza delle funzioni, le istruzioni preparate non vengono sovraccaricate in base al tipo o al numero di parametri specificati; il nome di un'istruzione preparata deve essere univoco all'interno di una sessione del database.

Quando viene emesso un comando EXECUTE per l'istruzione preparata, Amazon Redshift può facoltativamente rivedere il piano di esecuzione della query (per migliorare le prestazioni in base ai valori dei parametri specificati) prima di eseguire l'istruzione preparata. Inoltre, per ogni nuova esecuzione di un'istruzione preparata, Amazon Redshift può rivedere nuovamente il piano di esecuzione della query in base ai diversi valori dei parametri specificati con l'istruzione EXECUTE. Per esaminare il piano di esecuzione della query che Amazon Redshift ha scelto per qualsiasi istruzione EXECUTE specificata, utilizzare il comando EXPLAIN.

Per esempi e ulteriori informazioni sulla creazione e sull'uso delle istruzioni preparate, vedi PREPARE.

consultare anche

DEALLOCATE, PREPARE

EXPLAIN

Mostra il piano di esecuzione per l'istruzione di una query senza eseguire la query. Per informazioni sul flusso di lavoro di analisi delle query, consulta Flusso di lavoro dell'analisi di query.

Sintassi

```
EXPLAIN [ VERBOSE ] query
```

Parametri

VERBOSE

Visualizza il piano di query completo anziché solo un riepilogo.

query

Istruzione di query da spiegare. La query può essere un'istruzione SELECT, INSERT, CREATE TABLE AS, UPDATE o DELETE.

Note per l'utilizzo

Le prestazioni di EXPLAIN sono talvolta influenzate dal tempo necessario per creare le tabelle temporanee. Ad esempio, una query che utilizza l'ottimizzazione dell'espressione secondaria comune richiede che vengano create e analizzate le tabelle temporanee per restituire l'output di EXPLAIN. Il piano di query dipende dallo schema e dalle statistiche delle tabelle temporanee. Pertanto, il comando EXPLAIN per questo tipo di query potrebbe impiegare più tempo del previsto.

Puoi utilizzare EXPLAIN solo per i seguenti comandi:

- SELECT
- SELECT INTO
- CREATE TABLE AS
- INSERT
- UPDATE
- DELETE

Il comando EXPLAIN avrà esito negativo se lo si utilizza per altri comandi SQL, ad esempio operazioni di database o DDL (Data Definition Language).

I costi unitari relativi all'output EXPLAIN vengono utilizzati da Amazon Redshift per scegliere un piano di query. Amazon Redshift confronta le dimensioni delle varie stime delle risorse per determinare il piano.

Pianificazione di query e fasi di esecuzione

Il piano di esecuzione per una istruzione di query Amazon Redshift specifica suddivide l'esecuzione e il calcolo di una query in una sequenza discreta di fasi e operazioni di tabella che alla fine produce

un set di risultati finali per la query. Per informazioni sulla pianificazione di query, consultare Elaborazione query.

La seguente tabella fornisce un riepilogo delle fasi che Amazon Redshift può utilizzare nello sviluppo di un piano di esecuzione per qualsiasi query inviata dall'utente per l'esecuzione.

Operatori EXPLAIN	Fasi di esecuzione di query	Descrizione
SCAN:		
Scansione sequenziale	scan	Operatore o fase della scansione della relazione o della tabella di Amazon Redshift. Scansiona l'intera tabella in sequenza dall'iniz io alla fine; valuta anche i vincoli di query per ogni riga (Filtro) se specificato con la clausola WHERE. Usato anche per eseguire le Istruzioni INSERT, UPDATE e DELETE.
JOINS: Amazon Redshift usa differenti operatori di join secondo il progetto fisico delle tabelle combinate, la posizione dei dati necessari per il join e gli attributi specifici della stessa query. Scansione di sottoquery: la scansione e l'aggiunta delle sottoquery vengono utilizzate per eseguire query UNION.		
Loop nidificato	nloop	Il join meno ottimale, utilizzato principalmente per i cross join (prodotti cartesiani senza una condizione di join) e per alcuni join di disuguagl ianza.
Hash join	hjoin	Utilizzato anche per gli inner join e per gli outer join destra e sinistra e normalmente più veloce di un join loop nidificato. Hash Join legge la tabella esterna, applica l'hash alla colonna di join e trova corrispondenze nella tabella interna hash. La fase può essere versata sul disco. L'input interno di hjoin è una fase di hash che può essere basata su disco.

Operatori EXPLAIN	Fasi di esecuzione di query	Descrizione
Merge join	mjoin	Utilizzato anche per gli inner join e gli outer join (per le tabelle di join che sono sia distribuite che ordinate sulle colonne di unione). In genere il più veloce algoritmo join di Amazon Redshift, escludendo altre considerazioni sui costi.
AGGREGATION: operatori e fasi utilizzati per le query che implicano le funzioni aggregate e le operazioni GROUP BY.		
Aggregazione	aggr	Operatore/Fase per le funzioni aggregate scalari.
HashAggregate	aggr	Operatore/Fase per le funzioni aggregate raggruppate. Può operare dal disco in virtù del versamento della tabella di hash sul disco.
GroupAggregate	aggr	Operatore a volte scelto per le query aggregate raggruppate se l'impostazione di configura zione di Amazon Redshift per l'impostazione force_hash_grouping è disattivata.
SORT: operatori e fasi utilizzati quando le query devono ordinare o unire gli insiemi dei risultati.		
Ordina	sort	Sort esegue l'ordinamento specificato dalla clausola ORDER BY e altre operazioni come UNION e join. Può operare dal disco.
Unione	merge	Produce risultati ordinati finali di una query basata sui risultati ordinati intermedi che derivano dalle operazioni eseguite in parallelo.
Operazioni EXCEPT, INTERSECT e UNION:		
SetOp Tranne [Distinct]	hjoin	Utilizzato per eseguire le query EXCEPT. Può operare dal disco in virtù del fatto che l'hash di input può essere basato su disco.

Operatori EXPLAIN	Fasi di esecuzione di query	Descrizione
Hash Intersect [Distinct]	hjoin	Utilizzato per eseguire le query INTERSECT . Può operare dal disco in virtù del fatto che l'hash di input può essere basato su disco.
Append [All |Distinct]	save	Append usato con la scansione di sottoquery per implementare le query UNION e UNION ALL. Può operare dal disco in virtù di "save".
Varie/Altro:		
Hash	hash	Utilizzato per gli inner join e per gli outer join destra e sinistra (fornisce l'input a un join hash). L'operatore Hash crea la tabella hash per la tabella interna di un join. La tabella interna è la tabella che viene controllata per le corrispondenze e, in un join di due tabelle, è solitamente la più piccola delle due.
Limite	limit	Calcola la clausola LIMIT.
Materialize	save	Materializza le righe per l'input dei join di loop nidificati e alcuni merge join. Può operare dal disco.
--	parse	Utilizzato per analizzare i dati di input testuali durante un caricamento.
--	project	Utilizzato per riorganizzare le colonne e calcolare le espressioni, ovvero i dati del progetto.
Risultato	--	Esegue le funzioni scalari che non implicano alcun accesso di tabella.
--	return	Restituisce le righe al nodo principale o al client.

Operatori EXPLAIN	Fasi di esecuzione di query	Descrizione
Subplan	--	Utilizzato per determinate sottoquery.
Unique	unique	Elimina i duplicati dalle query SELECT DISTINCT e UNION.
Window	window	Calcola le funzioni di aggregazione e di classificazione della finestra. Può operare dal disco.
Operazioni di rete:		
Network (Broadcast)	bcast	Broadcast è anche un attributo di operatori e fasi Join Explain.
Network (Distribute)	dist	Distribuisce le righe ai nodi di calcolo per l'elaborazione parallela dal cluster del data warehouse.
Network (Send to Leader)	return	Invia risultati nuovamente al nodo principale per elaborazioni future.
Operazioni DML (operatori che modificano i dati):		
Insert (using Result)	insert	Inserisce i dati.
Delete (Scan + Filter)	Elimina	Elimina i dati. Può operare dal disco.
Update (Scan + Filter)	delete, insert	Implementato come delete e insert.

Utilizzo di EXPLAIN per RLS

Se una query contiene una tabella soggetta a politiche di sicurezza a livello di riga (RLS), EXPLAIN visualizza un nodo RLS speciale. SecureScan Amazon Redshift registra anche lo stesso tipo di nodo nella tabella di sistema STL_EXPLAIN. EXPLAIN non rivela il predicato RLS che si applica a dim_tbl. Il tipo di SecureScan nodo RLS serve da indicatore del fatto che il piano di esecuzione contiene operazioni aggiuntive invisibili all'utente corrente.

L'esempio seguente illustra un nodo RLS SecureScan .

```
EXPLAIN
SELECT D.cint
FROM fact_tbl F INNER JOIN dim_tbl D ON F.k_dim = D.k
WHERE F.k_dim / 10 > 0;
                              QUERY PLAN
------------------------------------------------------------------------
 XN Hash Join DS_DIST_ALL_NONE  (cost=0.08..0.25 rows=1 width=4)
   Hash Cond: ("outer".k_dim = "inner"."k")
   ->  *XN* *RLS SecureScan f  (cost=0.00..0.14 rows=2 width=4)*
         Filter: ((k_dim / 10) > 0)
   ->  XN Hash  (cost=0.07..0.07 rows=2 width=8)
         ->  XN Seq Scan on dim_tbl d  (cost=0.00..0.07 rows=2 width=8)
               Filter: (("k" / 10) > 0)
```

Per consentire un'analisi completa dei piani di query soggetti a RLS, Amazon Redshift offre le autorizzazioni di sistema EXPLAIN RLS. Gli utenti a cui è stata concessa questa autorizzazione possono esaminare piani di query completi che includono anche predicati RLS.

L'esempio seguente illustra un Seq Scan aggiuntivo sotto il SecureScan nodo RLS che include anche il predicato della policy RLS (k_dim > 1).

```
EXPLAIN SELECT D.cint
FROM fact_tbl F INNER JOIN dim_tbl D ON F.k_dim = D.k
WHERE F.k_dim / 10 > 0;
                                  QUERY PLAN
------------------------------------------------------------------------------
 XN Hash Join DS_DIST_ALL_NONE  (cost=0.08..0.25 rows=1 width=4)
   Hash Cond: ("outer".k_dim = "inner"."k")
   *->  XN RLS SecureScan f  (cost=0.00..0.14 rows=2 width=4)
         Filter: ((k_dim / 10) > 0)*
         ->  *XN* *Seq Scan on fact_tbl rls_table  (cost=0.00..0.06 rows=5 width=8)
               Filter: (k_dim > 1)*
   ->  XN Hash  (cost=0.07..0.07 rows=2 width=8)
         ->  XN Seq Scan on dim_tbl d  (cost=0.00..0.07 rows=2 width=8)
               Filter: (("k" / 10) > 0)
```

Sebbene l'autorizzazione EXPLAIN RLS sia concessa a un utente, Amazon Redshift registra il piano di query completo, inclusi i predicati RLS, nella tabella di sistema STL_EXPLAIN. Le query che vengono eseguite mentre questa autorizzazione non è concessa verranno registrate senza interni

RLS. La concessione o la rimozione dell'autorizzazione EXPLAIN RLS non modificherà ciò che Amazon Redshift ha registrato su STL_EXPLAIN per le query precedenti.

Relazioni Redshift protette da AWS Lake Formation-RLS

L'esempio seguente illustra un SecureScan nodo LF, che è possibile utilizzare per visualizzare le relazioni Lake Formation-RLS.

```
EXPLAIN
SELECT *
FROM lf_db.public.t_share
WHERE a > 1;
QUERY PLAN
------------------------------------------------------------
XN LF SecureScan t_share  (cost=0.00..0.02 rows=2 width=11)
(2 rows)
```

Esempi

> **Note**
>
> Per questi esempi, l'output di esempio potrebbe variare a seconda della configurazione di Amazon Redshift.

L'esempio seguente restituisce il piano di query per una query che seleziona EVENTID, EVENTNAME, VENUEID e VENUENAME dalle tabelle EVENT e VENUE:

```
explain
select eventid, eventname, event.venueid, venuename
from event, venue
where event.venueid = venue.venueid;
```

```
                              QUERY PLAN
------------------------------------------------------------------------
XN Hash Join DS_DIST_OUTER  (cost=2.52..58653620.93 rows=8712 width=43)
Hash Cond: ("outer".venueid = "inner".venueid)
->  XN Seq Scan on event  (cost=0.00..87.98 rows=8798 width=23)
->  XN Hash  (cost=2.02..2.02 rows=202 width=22)
->  XN Seq Scan on venue  (cost=0.00..2.02 rows=202 width=22)
```

```
(5 rows)
```

L'esempio seguente restituisce il piano di query per la stessa query con output dettagliato:

```
explain verbose
select eventid, eventname, event.venueid, venuename
from event, venue
where event.venueid = venue.venueid;
```

```
                                QUERY PLAN
------------------------------------------------------------------------------
{HASHJOIN
:startup_cost 2.52
:total_cost 58653620.93
:plan_rows 8712
:plan_width 43
:best_pathkeys <>
:dist_info DS_DIST_OUTER
:dist_info.dist_keys (
TARGETENTRY
{
VAR
:varno 2
:varattno 1
...

XN Hash Join DS_DIST_OUTER  (cost=2.52..58653620.93 rows=8712 width=43)
Hash Cond: ("outer".venueid = "inner".venueid)
->  XN Seq Scan on event  (cost=0.00..87.98 rows=8798 width=23)
->  XN Hash  (cost=2.02..2.02 rows=202 width=22)
->  XN Seq Scan on venue  (cost=0.00..2.02 rows=202 width=22)
(519 rows)
```

L'esempio seguente restituisce il piano di query per un'istruzione CREATE TABLE AS (CTAS):

```
explain create table venue_nonulls as
select * from venue
where venueseats is not null;

QUERY PLAN
-------------------------------------------------------------
XN Seq Scan on venue  (cost=0.00..2.02 rows=187 width=45)
```

```
Filter: (venueseats IS NOT NULL)
(2 rows)
```

FETCH

Recupera le righe usando un cursore. Per informazioni sulla dichiarazione di un cursore, vedi DECLARE.

FETCH recupera le righe in base alla posizione corrente nel cursore. Quando viene creato un cursore, viene posizionato prima della prima riga. Dopo un FETCH, il cursore viene posizionato sull'ultima riga recuperata. Se FETCH va oltre la fine delle righe disponibili, ad esempio a seguito di un FETCH ALL, il cursore viene lasciato posizionato dopo l'ultima riga.

FORWARD 0 recupera la riga corrente senza spostare il cursore, ossia recupera la riga recuperata più recentemente. Se il cursore è posizionato prima della prima riga o dopo l'ultima riga, non viene restituita alcuna riga.

Quando viene recuperata la prima riga di un cursore, l'intero set di risultati si materializza sul nodo principale, in memoria o sul disco, se necessario. A causa del potenziale impatto negativo sulle prestazioni dell'utilizzo dei cursori con set di risultati di grandi dimensioni, si consiglia di utilizzare approcci alternativi laddove possibile. Per ulteriori informazioni, consulta Considerazioni sulle prestazioni quando si utilizzano i cursori.

Per ulteriori informazioni, consultare DECLARE, CLOSE.

Sintassi

```
FETCH [ NEXT | ALL | {FORWARD [ count | ALL ] } ] FROM cursor
```

Parametri

NEXT

Recupera la riga successiva. Questa è l'impostazione predefinita.

ALL

Recupera tutte le righe rimanenti. Equivalente a FORWARD ALL. ALL non è supportato per i cluster a nodo singolo.

FORWARD [count | ALL]

Recupera le successive numero righe o tutte le righe rimanenti. `FORWARD 0` recupera la riga corrente. Per i cluster a nodo singolo, il valore massimo per il numero è `1000`. FORWARD ALL non è supportato per i cluster a nodo singolo.

cursor

Nome del nuovo cursore.

Esempio di FETCH

L'esempio seguente dichiara un cursore denominato LOLLAPALOOZA per selezionare le informazioni di vendita per l'evento Lollapalooza e quindi recupera le righe dal set di risultati utilizzando il cursore:

```
-- Begin a transaction

begin;

-- Declare a cursor

declare lollapalooza cursor for
select eventname, starttime, pricepaid/qtysold as costperticket, qtysold
from sales, event
where sales.eventid = event.eventid
and eventname='Lollapalooza';

-- Fetch the first 5 rows in the cursor lollapalooza:

fetch forward 5 from lollapalooza;

  eventname   |      starttime      | costperticket | qtysold
--------------+---------------------+---------------+---------
 Lollapalooza | 2008-05-01 19:00:00 |   92.00000000 |       3
 Lollapalooza | 2008-11-15 15:00:00 |  222.00000000 |       2
 Lollapalooza | 2008-04-17 15:00:00 |  239.00000000 |       3
 Lollapalooza | 2008-04-17 15:00:00 |  239.00000000 |       4
 Lollapalooza | 2008-04-17 15:00:00 |  239.00000000 |       1
(5 rows)

-- Fetch the next row:
```

```
fetch next from lollapalooza;

  eventname   |      starttime      | costperticket | qtysold
--------------+---------------------+---------------+---------
 Lollapalooza | 2008-10-06 14:00:00 |  114.00000000 |       2

-- Close the cursor and end the transaction:

close lollapalooza;
commit;
```

GRANT

Definisce le autorizzazioni di accesso per un utente o un ruolo.

Le autorizzazioni includono opzioni di accesso come la possibilità di leggere i dati in tabelle e viste, scrivere dati, creare ed eliminare tabelle. Utilizza questo comando per assegnare autorizzazioni specifiche per una tabella, un database, uno schema, una funzione, una procedura, una lingua o una colonna. Per revocare le autorizzazioni da un oggetto di database, utilizzare il comando REVOKE.

Le autorizzazioni includono anche le seguenti opzioni di accesso producer dell'unità di condivisione dati:

- Assegnazione dell'accesso dell'unità di condivisione dati agli spazi dei nomi e agli account consumer.
- Assegnazione dell'autorizzazione a modificare un'unità di condivisione dati aggiungendo o rimuovendo oggetti dell'unità di condivisione dati.
- Assegnazione dell'autorizzazione a condividere un'unità di condivisione dati aggiungendo o rimuovendo spazi dei nomi consumer dell'unità di condivisione dati.

Le opzioni di accesso consumer dell'unità di condivisione dati sono le seguenti:

- Assegnazione dell'accesso completo degli utenti ai database creati da un'unità di condivisione dati o a schemi esterni che fanno riferimento a tali database.
- Assegnazione delle autorizzazioni degli utenti a livello di oggetto per i database creati da un'unità di condivisione dati, come avviene per gli oggetti di database locali. Per assegnare questo livello di autorizzazione, è necessario utilizzare la clausola WITH PERMISSIONS durante la creazione di un database dall'unità di condivisione dati. Per ulteriori informazioni, consulta CREATE DATABASE.

Per ulteriori informazioni sulle autorizzazioni dell'unità di condivisione dati, consulta Condivisione di unità di condivisione dati.

È anche possibile assegnare ruoli per gestire le autorizzazioni del database e controllare cosa possono fare gli utenti in relazione ai propri dati. Definendo i ruoli e assegnando ruoli agli utenti, è possibile limitare le operazioni che tali utenti possono intraprendere, ad esempio limitando gli utenti ai soli comandi CREATE TABLE e INSERT. Per ulteriori informazioni sul comando CREATE ROLE, consulta the section called “CREATE ROLE”. Amazon Redshift ha alcuni ruoli definiti dal sistema che è possibile utilizzare anche per concedere autorizzazioni specifiche ai propri utenti. Per ulteriori informazioni, consulta the section called “Ruoli definiti dal sistema di Amazon Redshift”.

Per quanto riguarda e autorizzazioni, è possibile solo GRANT o REVOKE USAGE in uno schema esterno per gli utenti di database e gruppi di utenti che utilizzano la sintassi ON SCHEMA. Quando si utilizza ON EXTERNAL SCHEMA con AWS Lake Formation, è possibile concedere e revocare solo le autorizzazioni a un ruolo (IAM). AWS Identity and Access Management Per un elenco delle autorizzazioni richieste, consulta la sintassi.

Per le procedure archiviate, l'unica autorizzazione che si può concedere è EXECUTE.

Non è possibile eseguire GRANT (su una risorsa esterna) all'interno di un blocco di transazione (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Per vedere quali autorizzazioni sono state concesse agli utenti per un database, usa HAS_DATABASE_PRIVILEGE. Per vedere quali autorizzazioni sono state concesse agli utenti per uno schema, usa PRIVILEGIO DELLO SCHEMA HAS. Per vedere quali autorizzazioni sono state concesse agli utenti per una tabella, usa HAS_TABLE_PRIVILEGE.

Sintassi

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | DROP | REFERENCES | ALTER | TRUNCATE }
 [,...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...] | ALL TABLES IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]

GRANT { { CREATE | TEMPORARY | TEMP | ALTER } [,...] | ALL [ PRIVILEGES ] }
    ON DATABASE db_name [, ...]
    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]
```

```
GRANT { { CREATE | USAGE | ALTER } [,...] | ALL [ PRIVILEGES ] }
    ON SCHEMA schema_name [, ...]
    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
    ON { FUNCTION function_name ( [ [ argname ] argtype [, ...] ] ) [, ...] | ALL
 FUNCTIONS IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
    ON { PROCEDURE procedure_name ( [ [ argname ] argtype [, ...] ] ) [, ...] | ALL
 PROCEDURES IN SCHEMA schema_name [, ...] }
    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]

GRANT USAGE
    ON LANGUAGE language_name [, ...]
    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]
```

Concessione di autorizzazioni a livello di colonna per le tabelle

Di seguito è riportata la sintassi per le autorizzazioni a livello di colonna su tabelle e viste di Amazon Redshift.

```
GRANT { { SELECT | UPDATE } ( column_name [, ...] ) [, ...] | ALL [ PRIVILEGES ]
 ( column_name [,...] ) }
     ON { [ TABLE ] table_name [, ...] }

     TO { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
```

Concessione delle autorizzazioni ASSUMEROLE

Di seguito è riportata la sintassi per le autorizzazioni ASSUMEROLE concesse a utenti e gruppi con un ruolo specificato. Per iniziare a utilizzare il privilegio ASSUMEROLE, consultare Note di utilizzo per la concessione dell'autorizzazione ASSUMEROLE.

```
GRANT ASSUMEROLE
       ON { 'iam_role' [, ...] | default | ALL }
       TO { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
```

```
        FOR { ALL | COPY | UNLOAD | EXTERNAL FUNCTION | CREATE MODEL } [, ...]
```

Concessione di autorizzazioni per l'integrazione di Redshift Spectrum con Lake Formation

Di seguito è riportata la sintassi per l'integrazione di Redshift Spectrum con Lake Formation.

```
GRANT { SELECT | ALL [ PRIVILEGES ] } ( column_list )
    ON EXTERNAL TABLE schema_name.table_name
    TO { IAM_ROLE iam_role } [, ...] [ WITH GRANT OPTION ]

GRANT { { SELECT | ALTER | DROP | DELETE | INSERT }  [, ...] | ALL [ PRIVILEGES ] }
    ON EXTERNAL TABLE schema_name.table_name [, ...]
    TO { { IAM_ROLE iam_role } [, ...] | PUBLIC } [ WITH GRANT OPTION ]

GRANT { { CREATE | ALTER | DROP }  [, ...] | ALL [ PRIVILEGES ] }
    ON EXTERNAL SCHEMA schema_name [, ...]
    TO { IAM_ROLE iam_role } [, ...] [ WITH GRANT OPTION ]
```

Concessione delle autorizzazioni per l'unità di condivisione dati

Autorizzazioni dell'unità di condivisione dati sul lato producer

Di seguito è riportata la sintassi per l'utilizzo di GRANT per assegnare le autorizzazioni ALTER o SHARE a un utente o un ruolo. L'utente può modificare l'unità di condivisione dati con l'autorizzazione ALTER o assegnare l'utilizzo a un consumer con l'autorizzazione SHARE. ALTER e SHARE sono le uniche autorizzazioni che è possibile assegnare a utenti e gruppi di utenti per un'unità di condivisione dati.

```
GRANT { ALTER | SHARE } ON DATASHARE datashare_name
    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]
```

Di seguito è riportata la sintassi per l'utilizzo di GRANT per le autorizzazioni per l'utilizzo dell'unità di condivisione dati su Amazon Redshift. Si concede l'accesso a una unità di condivisione dati a un consumer utilizzando l'autorizzazione USAGE. Non è possibile concedere questa autorizzazione agli utenti o ai gruppi di utenti. Questa autorizzazione inoltre non supporta WITH GRANT OPTION per l'istruzione GRANT. Solo gli utenti o i gruppi di utenti con autorizzazione SHARE precedentemente concessa loro PER l'unità di condivisione dati possono eseguire questo tipo di istruzione GRANT.

```
GRANT USAGE
    ON DATASHARE datashare_name
```

```
    TO NAMESPACE 'namespaceGUID' | ACCOUNT 'accountnumber' [ VIA DATA CATALOG ]
```

Di seguito è riportato un esempio di come concedere l'autorizzazione per l'utilizzo di un'unità di condivisione dati a un account Lake Formation.

```
GRANT USAGE ON DATASHARE salesshare TO ACCOUNT '123456789012' VIA DATA CATALOG;
```

Autorizzazioni dell'unità di condivisione dati sul lato consumer

Di seguito è riportata la sintassi per le autorizzazioni di utilizzo per la condivisione dei dati GRANT per un database o uno schema specifico creato da una unità di condivisione dati.

Le ulteriori autorizzazioni richieste per l'accesso dei consumer a un database creato da un'unità di condivisione dati variano a seconda che il comando CREATE DATABASE utilizzato per creare il database dall'unità di condivisione dati includa o meno la clausola WITH PERMISSIONS. Per ulteriori informazioni sul comando CREATE DATABASE con la clausola WITH PERMISSIONS, consulta CREATE DATABASE.

Database creati senza la clausola WITH PERMISSIONS

Quando assegni l'autorizzazione USAGE per un database creato da un'unità di condivisione dati senza la clausola WITH PERMISSIONS, non è necessario fornire separatamente le autorizzazioni per gli oggetti nel database condiviso. Le entità a cui è fornito l'utilizzo dei database creati dall'unità di condivisione dati senza la clausola WITH PERMISSIONS hanno automaticamente accesso a tutti gli oggetti del database.

Database creati con la clausola WITH PERMISSIONS

Quando assegni USAGE per un database condiviso che è stato creato da un'unità di condivisione dati con la clausola WITH PERMISSIONS, alle identità lato consumer devono comunque essere fornite le autorizzazioni pertinenti per gli oggetti del database condiviso per potervi accedere, proprio come si farebbe per le autorizzazioni degli oggetti del database locale. Per assegnare le autorizzazioni agli oggetti di un database creato da un'unità di condivisione dati, utilizza la sintassi in tre parti `database_name.schema_name.object_name`. Per assegnare le autorizzazioni agli oggetti in uno schema esterno che fa riferimento a uno schema condiviso del database condiviso, utilizza la sintassi in due parti `schema_name.object_name`.

```
GRANT USAGE ON { DATABASE shared_database_name [, ...] | SCHEMA shared_schema}
    TO { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
```

Assegnazione delle autorizzazioni con ambito

Le autorizzazioni con ambito consentono di concedere autorizzazioni a un utente o a un ruolo su tutti gli oggetti di un tipo all'interno di un database o di uno schema. Gli utenti e i ruoli con autorizzazioni mirate dispongono delle autorizzazioni specificate su tutti gli oggetti attuali e futuri all'interno del database o dello schema.

Di seguito è riportata la sintassi per assegnare a utenti e ruoli le autorizzazioni con ambito. Per ulteriori informazioni sulle autorizzazioni con ambito, vedere. Autorizzazioni con ambito

```
GRANT { CREATE | USAGE | ALTER } [,...] | ALL [ PRIVILEGES ] }
FOR SCHEMAS IN
DATABASE db_name
TO { username [ WITH GRANT OPTION ] | ROLE role_name } [, ...]

GRANT
{ { SELECT | INSERT | UPDATE | DELETE | DROP | ALTER | TRUNCATE | REFERENCES }
 [, ...] } | ALL [PRIVILEGES] } }
FOR TABLES IN
{SCHEMA schema_name [DATABASE db_name ] | DATABASE db_name }
TO { username [ WITH GRANT OPTION ] | ROLE role_name} [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
FOR FUNCTIONS IN
{SCHEMA schema_name [DATABASE db_name ] | DATABASE db_name }
TO { username [ WITH GRANT OPTION ] | ROLE role_name | } [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
FOR PROCEDURES IN
{SCHEMA schema_name [DATABASE db_name ] | DATABASE db_name }
TO { username [ WITH GRANT OPTION ] | ROLE role_name | } [, ...]

GRANT USAGE
FOR LANGUAGES IN
{DATABASE db_name}
TO { username [ WITH GRANT OPTION ] | ROLE role_name } [, ...]
```

Tieni presente che le autorizzazioni con ambito non fanno distinzione tra le autorizzazioni per le funzioni e per le procedure. Ad esempio, l'istruzione seguente concede l'`EXECUTE`autorizzazione sia per bob le funzioni che per le procedure dello schema. `Sales_schema`

```
GRANT EXECUTE FOR FUNCTIONS IN SCHEMA Sales_schema TO bob;
```

Concessione delle autorizzazioni di machine learning

Di seguito è riportata la sintassi per le autorizzazioni per il modello di machine learning su Amazon Redshift.

```
GRANT CREATE MODEL
    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]

GRANT { EXECUTE | ALL [ PRIVILEGES ] }
    ON MODEL model_name [, ...]

    TO { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]
```

Concessione delle autorizzazioni per i ruoli

Di seguito è riportata la sintassi per concedere le autorizzazioni per i ruoli su Amazon Redshift.

```
GRANT { ROLE role_name } [, ...] TO { { user_name [ WITH ADMIN OPTION ] } |
 ROLE role_name }[, ...]
```

Di seguito è riportata la sintassi per concedere le autorizzazioni di sistema per i ruoli su Amazon Redshift.

```
GRANT
  {
    { CREATE USER | DROP USER | ALTER USER |
    CREATE SCHEMA | DROP SCHEMA |
    ALTER DEFAULT PRIVILEGES |
    ACCESS CATALOG |
    CREATE TABLE | DROP TABLE | ALTER TABLE |
    CREATE OR REPLACE FUNCTION | CREATE OR REPLACE EXTERNAL FUNCTION |
    DROP FUNCTION |
    CREATE OR REPLACE PROCEDURE | DROP PROCEDURE |
    CREATE OR REPLACE VIEW | DROP VIEW |
    CREATE MODEL | DROP MODEL |
    CREATE DATASHARE | ALTER DATASHARE | DROP DATASHARE |
    CREATE LIBRARY | DROP LIBRARY |
    CREATE ROLE | DROP ROLE |
    TRUNCATE TABLE
    VACUUM | ANALYZE | CANCEL }[, ...]
```

```
  }
  | { ALL [ PRIVILEGES ] }
TO { ROLE role_name } [, ...]
```

Concessione di autorizzazioni esplicative per i filtri delle policy di sicurezza a livello di riga

Di seguito è riportata la sintassi per la concessione delle autorizzazioni che spiega i filtri delle policy di sicurezza a livello di riga di una query nel piano EXPLAIN. Puoi revocare il privilegio utilizzando l'istruzione REVOKE.

```
GRANT EXPLAIN RLS TO ROLE rolename
```

Di seguito è riportata la sintassi per concedere le autorizzazioni per aggirare le policy di sicurezza a livello di riga per una query.

```
GRANT IGNORE RLS TO ROLE rolename
```

Concessione delle autorizzazioni per le tabelle di ricerca RLS a un oggetto policy

Di seguito è riportata la sintassi per concedere le autorizzazioni alla policy di sicurezza a livello di riga.

```
GRANT SELECT ON [ TABLE ] table_name [, ...]
TO RLS POLICY policy_name [, ...]
```

Parametri

SELECT

Revoca l'autorizzazione a selezionare i dati da una tabella o una vista utilizzando un'istruzione SELECT. L'autorizzazione SELECT è richiesta anche per fare riferimento ai valori di colonna esistenti per le operazioni UPDATE o DELETE.

INSERT

Concede l'autorizzazione a caricare i dati in una tabella utilizzando un'istruzione INSERT o un'istruzione COPY.

UPDATE

Concede l'autorizzazione ad aggiornare una colonna della tabella utilizzando un'istruzione UPDATE. Le operazioni UPDATE richiedono anche l'autorizzazione SELECT, perché devono fare

riferimento alle colonne della tabella per determinare quali righe aggiornare o per calcolare nuovi valori per le colonne.

DELETE

Concede l'autorizzazione a eliminare una riga di dati da una tabella. Le operazioni DELETE richiedono anche il privilegio SELECT, perché devono fare riferimento alle colonne della tabella per determinare quali righe eliminare.

DROP

Concede l'autorizzazione a eliminare una tabella. Questa autorizzazione si applica in Amazon Redshift e in un AWS Glue Data Catalog ambiente abilitato per Lake Formation.

REFERENCES

Concede l'autorizzazione a creare un vincolo di chiave esterna. È necessario concedere questa autorizzazione sia sulla tabella a cui si fa riferimento sia sulla tabella che fa riferimento; in caso contrario, l'utente non può creare il vincolo.

ALTER

A seconda dell'oggetto del database, concede le seguenti autorizzazioni all'utente o al gruppo di utenti:

- Per le tabelle, ALTER concede l'autorizzazione a modificare una tabella o una vista. Per ulteriori informazioni, consulta ALTER TABLE.
- Per i database, ALTER concede l'autorizzazione a modificare un database. Per ulteriori informazioni, consulta ALTER DATABASE.
- Per gli schemi, ALTER concede l'autorizzazione a modificare uno schema. Per ulteriori informazioni, consulta ALTER SCHEMA.
- Per le tabelle esterne, ALTER concede l'autorizzazione a modificare una tabella in un file AWS Glue Data Catalog abilitato per Lake Formation. Questa autorizzazione si applica solo quando si utilizza Lake Formation.

TRUNCATE

Concede l'autorizzazione a troncare una tabella. Senza questa autorizzazione, solo il proprietario di una tabella o un utente con privilegi avanzati può troncare una tabella. Per ulteriori informazioni sul comando TRUNCATE, consulta the section called “TRUNCATE”.

ALL [PRIVILEGES]

Concede tutte le autorizzazioni disponibili contemporaneamente all'utente o al gruppo di utenti specificato. La parola chiave PRIVILEGES è facoltativa.

GRANT ALL ON SCHEMA non concede le autorizzazioni CREATE per gli schemi esterni.

Puoi concedere l'autorizzazione ALL a una tabella in un AWS Glue Data Catalog che è abilitata per Lake Formation. In questo caso, le singole autorizzazioni (come SELECT, ALTER e così via) sono registrate nel catalogo dati.

ASSUMEROLE

Concede l'autorizzazione a eseguire i comandi COPY, UNLOAD, EXTERNAL FUNCTION e CREATE MODEL a utenti, ruoli o gruppi con un ruolo specificato. L'utente, il ruolo o il gruppo assume tale ruolo all'esecuzione del comando specificato. Per iniziare a utilizzare l'autorizzazione ASSUMEROLE, consultare Note di utilizzo per la concessione dell'autorizzazione ASSUMEROLE.

ON [TABLE] table_name

Concede le autorizzazioni specificate su una tabella o una vista. La parola chiave TABLE è facoltativa. Puoi elencare più tabelle e viste in un'unica istruzione.

ON ALL TABLES IN SCHEMA schema_name

Concede le autorizzazioni specificate su tutte le tabelle e le viste nello schema a cui si fa riferimento.

(column_name [,...]) ON TABLE table_name

Concede le autorizzazioni specificate a utenti, gruppi o PUBLIC sulle colonne specificate della tabella o della vista di Amazon Redshift.

(column_list) ON EXTERNAL TABLE schema_name.table_name

Concede le autorizzazioni specificate a un ruolo IAM nelle colonne indicate della tabella Lake Formation nello schema a cui si fa riferimento.

ON EXTERNAL TABLE schema_name.table_name

Concede le autorizzazioni specificate a un ruolo IAM nelle tabelle Lake Formation indicate dello schema a cui si fa riferimento.

ON EXTERNAL SCHEMA schema_name

Concede le autorizzazioni specificate a un ruolo IAM nello schema a cui si fa riferimento.

ON ruolo_iam

Concede le autorizzazioni specificate a un ruolo IAM.

TO username

Indica l'utente che riceve le autorizzazioni.

TO IAM_ROLE iam_role

Indica il ruolo IAM che riceve le autorizzazioni.

WITH GRANT OPTION

Indica che l'utente che riceve le autorizzazioni può a sua volta concedere le stesse autorizzazioni ad altri. WITH GRANT OPTION non può essere concesso a un gruppo o a PUBLIC.

ROLE role_name

Concede le autorizzazioni a un ruolo.

GROUP group_name

Concede le autorizzazioni a un gruppo di utenti. Può essere un elenco separato da virgole per specificare più gruppi di utenti.

PUBLIC

Concede le autorizzazioni specificate a tutti gli utenti, inclusi gli utenti creati in un momento successivo. PUBLIC rappresenta un gruppo che include sempre tutti gli utenti. Le autorizzazioni di un singolo utente consistono nella somma delle autorizzazioni concesse a PUBLIC, delle autorizzazioni concesse a tutti i gruppi a cui l'utente appartiene e di tutte le autorizzazioni concesse all'utente singolarmente.

La concessione di PUBLIC a una EXTERNAL TABLE di Lake Formation implica la concessione dell'autorizzazione al gruppo everyone di Lake Formation.

CREATE

A seconda dell'oggetto del database, concede le seguenti autorizzazioni all'utente o al gruppo di utenti:

- Per i database, CREATE consente agli utenti di creare schemi all'interno del database.
- Per gli schemi, CREATE consente agli utenti di creare oggetti all'interno di uno schema. Per rinominare un oggetto, l'utente deve avere l'autorizzazione CREATE ed essere il proprietario dell'oggetto da rinominare.

- CREATE ON SCHEMA non è supportato per gli schemi esterni di Amazon Redshift Spectrum. Per garantire l'utilizzo di tabelle esterne in uno schema esterno, concedi USAGE ON SCHEMA agli utenti che devono accedere. Solo il proprietario di uno schema esterno o di un utente con privilegi avanzati è autorizzato a creare tabelle esterne nello schema esterno. Per trasferire la proprietà di uno schema esterno, utilizza ALTER SCHEMA per cambiare il proprietario.

TEMPORARY | TEMP

Concede l'autorizzazione a creare tabelle temporanee nel database specificato. Per eseguire query Amazon Redshift Spectrum, l'utente del database deve avere l'autorizzazione per creare tabelle temporanee nel database.

> **Note**
>
> Per impostazione predefinita, agli utenti è concessa l'autorizzazione di creare tabelle temporanee tramite la loro appartenenza automatica al gruppo PUBLIC. Per rimuovere l'autorizzazione per gli utenti di creare tabelle temporanee, revoca l'autorizzazione TEMP dal gruppo PUBLIC. Quindi concedere esplicitamente l'autorizzazione per creare tabelle temporanee per utenti o gruppi di utenti specifici.

ON DATABASE db_name

Concede le autorizzazioni specificate su un database.

USAGE

Concede l'autorizzazione USAGE su uno schema specifico che rende gli oggetti in tale schema accessibili agli utenti. Le autorizzazioni per operazioni specifiche su questi oggetti devono essere concesse separatamente (ad esempio, le autorizzazioni SELECT o UPDATE sulle tabelle) per gli schemi Amazon Redshift locali. Per impostazione predefinita, tutti gli utenti dispongono delle autorizzazioni CREATE e USAGE per lo schema PUBLIC.

Quando si concede l'autorizzazione USAGE a schemi esterni utilizzando la sintassi ON SCHEMA, non è necessario concedere autorizzazioni per azioni separatamente sugli oggetti nello schema esterno. Le autorizzazioni del catalogo corrispondenti controllano le autorizzazioni granulari per gli oggetti dello schema esterno.

ON SCHEMA schema_name

Concede le autorizzazioni specificate su uno schema.

GRANT CREATE ON SCHEMA e l'autorizzazione CREATE in GRANT ALL ON SCHEMA non sono supportati per gli schemi esterni di Amazon Redshift Spectrum. Per garantire l'utilizzo di tabelle esterne in uno schema esterno, concedi USAGE ON SCHEMA agli utenti che devono accedere. Solo il proprietario di uno schema esterno o di un utente con privilegi avanzati è autorizzato a creare tabelle esterne nello schema esterno. Per trasferire la proprietà di uno schema esterno, utilizza ALTER SCHEMA per cambiare il proprietario.

EXECUTE ON ALL FUNCTIONS IN SCHEMA schema_name

Concede le autorizzazioni specificate su tutte le funzioni nello schema a cui si fa riferimento.

Amazon Redshift non supporta le istruzioni GRANT o REVOKE per le voci integrate pg_proc definite nello spazio dei nomi pg_catalog.

EXECUTE ON PROCEDURE procedure_name

Concede l'autorizzazione EXECUTE su una procedura archiviata specifica. Poiché i nomi delle procedure archiviate possono essere in overload, devi includere l'elenco degli argomenti per la procedura. Per ulteriori informazioni, consulta Denominazione delle stored procedure.

EXECUTE ON ALL PROCEDURES IN SCHEMA schema_name

Concede le autorizzazione specificate su tutte le procedure archiviate nello schema a cui si fa riferimento.

USAGE ON LANGUAGE language_name

Concede l'autorizzazione USAGE per una lingua.

L'autorizzazione USAGE ON LANGUAGE è necessaria per creare funzioni definite dall'utente (UDF) eseguendo il comando CREATE FUNCTION. Per ulteriori informazioni, consulta Sicurezza e privilegi dell'UDF.

È necessaria l'autorizzazione USAGE ON LANGUAGE per creare procedure archiviate eseguendo il comando CREATE PROCEDURE. Per ulteriori informazioni, consulta Sicurezza e privilegi per le procedure archiviate .

Per le funzioni definite dall'utente Python usa `plpythonu`. Per le funzioni definite dall'utente SQL usa `sql`. Per le procedure archiviate, usa `plpgsql`.

FOR { ALL | COPY | UNLOAD | EXTERNAL FUNCTION | CREATE MODEL } [, ...]

Specifica il comando SQL per il quale viene concessa l'autorizzazione. Puoi specificare ALL per concedere l'autorizzazione sulle istruzioni COPY, UNLOAD, EXTERNAL FUNCTION e CREATE MODEL. Questa clausola si applica solo alla concessione dell'autorizzazione ASSUMEROLE.

ALTER

Concede l'autorizzazione ALTER agli utenti per aggiungere o rimuovere oggetti da una unità di condivisione dati o per impostare la proprietà PUBLICACCESSIBLE. Per ulteriori informazioni, consulta ALTER DATASHARE.

SHARE

Concede autorizzazioni a utenti e gruppi di utenti per aggiungere consumer di dati a una unità di condivisione dati. Questa autorizzazione è necessaria per consentire al particolare consumer (account o spazio dei nomi) di accedere all'unità di condivisione dati dai propri cluster. Il consumatore può essere lo stesso account o un AWS account diverso, con lo stesso spazio dei nomi del cluster o uno diverso, come specificato da un identificatore univoco globale (GUID).

ON DATASHARE nome_unità_condivisione_dati

Concede le autorizzazioni specificate sull'unità di condivisione dati a cui si fa riferimento. Per informazioni sulla granularità del controllo degli accessi dei consumer, consultare Condivisione di dati a diversi livelli in Amazon Redshift.

USAGE

Quando USAGE viene concesso a un account consumer o a uno spazio dei nomi all'interno dello stesso account, l'account consumer specifico o lo spazio dei nomi all'interno dell'account potrà accedere all'unità di condivisione dati e gli oggetti dell'unità di condivisione dati saranno in sola lettura.

TO NAMESPACE 'clusternamespace GUID'

Indica uno spazio dei nomi nello stesso account in cui i consumer possono ricevere le autorizzazioni specificate per l'unità di condivisione dati. Gli spazi dei nomi utilizzano un GUID alfanumerico a 128 bit.

TO ACCOUNT 'accountnumber' [VIA DATA CATALOG]

Indica il numero di un altro account i cui consumer possono ricevere le autorizzazioni specificate per l'unità di condivisione dati. Quando si specifica "VIA DATA CATALOG" si sta concedendo l'autorizzazione per l'utilizzo dell'unità di condivisione dati a un account Lake Formation.

L'omissione di questo parametro indica che si sta concedendo l'autorizzazione per l'utilizzo a un account proprietario del cluster.

ON DATABASE nome_database_condiviso> [, ...]

Concede le autorizzazioni di utilizzo per il database specificato creato nell'unità di condivisione dati specificata.

ON SCHEMA schema_condiviso

Concede le autorizzazioni per lo schema specificato creato nell'unità di condivisione dati specificata.

FOR { SCHEMAS | TABLES | FUNCTIONS | PROCEDURES | LANGUAGES } IN

Specifica gli oggetti del database a cui assegnare l'autorizzazione. I parametri successivi a IN definiscono l'ambito dell'autorizzazione assegnata.

CREATE MODEL

Concede l'autorizzazione CREATE MODEL a utenti o gruppi di utenti specifici.

ON MODEL nome_modello

Concede l'autorizzazione EXECUTE su un modello specifico.

ACCESS CATALOG

Assegna l'autorizzazione a visualizzare i metadati pertinenti degli oggetti a cui il ruolo ha accesso.

{ role } [, ...]

Il ruolo da concedere a un altro ruolo, un utente o PUBLIC.

PUBLIC rappresenta un gruppo che include sempre tutti gli utenti. Le autorizzazioni di un singolo utente consistono nella somma delle autorizzazioni concesse a PUBLIC, delle autorizzazioni concesse a tutti i gruppi a cui l'utente appartiene e di tutte le autorizzazioni concesse all'utente singolarmente.

TO { { user_name [WITH ADMIN OPTION] } | role }[, ...]

Concede il ruolo specificato a un utente specificato con l'opzione WITH ADMIN OPTION, un altro ruolo o PUBLIC.

La clausola WITH ADMIN OPTION fornisce le opzioni di amministrazione per tutti i ruoli concessi a tutti gli assegnatari.

EXPLAIN RLS TO ROLE rolename

Concede a un ruolo l'autorizzazione a spiegare i filtri delle policy di sicurezza a livello di riga di una query nel piano EXPLAIN.

IGNORE RLS TO ROLE rolename

Concede a un ruolo l'autorizzazione a escludere le policy di sicurezza a livello di riga per una query.

Note per l'utilizzo

Per ulteriori informazioni sulle note di utilizzo di GRANT, consulta the section called “Note per l'utilizzo”.

Esempi

Per gli esempi di come utilizzare GRANT, consulta the section called “Esempi”.

Note per l'utilizzo

Per concedere i privilegi su un oggetto, è necessario soddisfare uno dei seguenti criteri:

- Essere il proprietario dell'oggetto.
- Essere un utente con privilegi avanzati.
- Avere un privilegio di concessione per l'oggetto e il privilegio.

Ad esempio, il seguente comando consente all'utente HR di eseguire i comandi SELECT sulla tabella dei dipendenti e di concedere e revocare lo stesso privilegio per altri utenti.

```
grant select on table employees to HR with grant option;
```

HR non può concedere privilegi per operazioni diverse da SELECT o su qualsiasi altra tabella rispetto a quella dei dipendenti.

Ad esempio, il seguente comando consente all'utente HR di eseguire i comandi ALTER sulla tabella dei dipendenti e di concedere e revocare lo stesso privilegio per altri utenti.

```
grant ALTER on table employees to HR with grant option;
```

HR non può concedere privilegi per operazioni diverse da ALTER o su qualsiasi altra tabella che non sia quella dei dipendenti.

Avere i privilegi su una vista non implica i privilegi sulle tabelle sottostanti. Allo stesso modo, avere i privilegi su uno schema non implica i privilegi sulle tabelle dello schema. Invece, è necessario concedere l'accesso alle tabelle sottostanti in modo esplicito.

Per concedere i privilegi a una AWS Lake Formation tabella, il ruolo IAM associato allo schema esterno della tabella deve disporre dell'autorizzazione a concedere i privilegi alla tabella esterna. L'esempio seguente crea uno schema esterno con un ruolo IAM associato `myGrantor`. Il ruolo IAM `myGrantor` ha l'autorizzazione per concedere le autorizzazioni ad altri. Il comando GRANT utilizza l'autorizzazione del ruolo IAM `myGrantor` associato allo schema esterno per concedere l'autorizzazione al ruolo IAM `myGrantee`.

```
create external schema mySchema
from data catalog
database 'spectrum_db'
iam_role 'arn:aws:iam::123456789012:role/myGrantor'
create external database if not exists;
```

```
grant select
on external table mySchema.mytable
to iam_role 'arn:aws:iam::123456789012:role/myGrantee';
```

Se vengono concessi i privilegi GRANT ALL a un ruolo IAM, i singoli privilegi vengono concessi nel relativo catalogo dati abilitato per Lake Formation. Ad esempio, il seguente GRANT ALL comporta la concessione di privilegi singoli (SELECT, ALTER, DROP, DELETE e INSERT) nella console Lake Formation.

```
grant all
on external table mySchema.mytable
to iam_role 'arn:aws:iam::123456789012:role/myGrantee';
```

Gli utenti con privilegi avanzati possono accedere a tutti gli oggetti indipendentemente dai comandi GRANT e REVOKE che impostano i privilegi dell'oggetto.

Note di utilizzo per il controllo degli accessi a livello di colonna

Le seguenti note di utilizzo si applicano ai privilegi a livello di colonna su tabelle e viste di Amazon Redshift. Queste note descrivono le tabelle; le stesse note si applicano alle viste a meno che non si noti esplicitamente un'eccezione.

- Per una tabella Amazon Redshift, è possibile concedere i privilegi SELECT e UPDATE solo a livello di colonna. Per una vista Amazon Redshift, è possibile concedere solo il privilegio SELECT a livello di colonna.
- La parola chiave ALL è un sinonimo di privilegi SELECT e UPDATE combinati quando utilizzata nel contesto di una GRANT a livello di colonna in una tabella.
- Se non disponi del privilegio SELECT su tutte le colonne di una tabella, l'esecuzione di un'operazione SELECT * restituisce solo le colonne a cui hai accesso. Quando si utilizza una vista, un'operazione SELECT * tenta di accedere a tutte le colonne della vista. Se non si dispone dell'autorizzazione per accedere a tutte le colonne, queste query hanno esito negativo e viene visualizzato un errore di autorizzazione negata.
- SELECT * non si espande solo alle colonne accessibili nei seguenti casi:
  - Non è possibile creare una vista normale con solo colonne accessibili utilizzando SELECT *.
  - Non è possibile creare una vista materializzata con solo colonne accessibili utilizzando SELECT *.
- Se si dispone di privilegi SELECT o UPDATE su una tabella o vista e si aggiunge una colonna, si hanno ancora gli stessi privilegi sulla tabella o sulla vista e quindi tutte le relative colonne.
- Solo il proprietario di una tabella o un utente con privilegi avanzati possono concedere privilegi a livello di colonna.
- La clausola WITH GRANT OPTION non è supportata per i privilegi a livello di colonna.
- Non è possibile mantenere lo stesso privilegio sia a livello di tabella sia a livello di colonna. Ad esempio, l'utente `data_scientist` non può avere sia il privilegio SELECT nella tabella `employee` sia il privilegio SELECT nella colonna `employee.department`. Considerare i seguenti risultati quando si concede lo stesso privilegio a una tabella e a una colonna all'interno della tabella:
  - Se un utente dispone di un privilegio a livello di tabella su una tabella, la concessione dello stesso privilegio a livello di colonna non ha alcun effetto.
  - Se un utente dispone di un privilegio a livello di tabella in una tabella, la revoca dello stesso privilegio per una o più colonne della tabella restituisce un errore. Al contrario, revoca il privilegio a livello di tabella.

  - Se un utente dispone di un privilegio a livello di colonna, la concessione dello stesso privilegio a livello di tabella restituisce un errore.
  - Se un utente dispone di un privilegio a livello di colonna, la revoca dello stesso privilegio a livello di tabella revoca sia i privilegi di colonna sia di tabella per tutte le colonne della tabella.
- Non è possibile concedere privilegi a livello di colonna nelle viste con associazione tardiva.
- Per creare una vista materializzata, è necessario disporre del privilegio SELECT a livello di tabella nelle tabelle di base. Anche se si dispone di privilegi a livello di colonna su colonne specifiche, non è possibile creare una vista materializzata solo su tali colonne. Tuttavia, è possibile concedere il privilegio SELECT alle colonne di una vista materializzata, in modo simile alle viste normali.
- Per cercare le concessioni dei privilegi a livello di colonna, utilizzare la vista PG_ATTRIBUTE_INFO .

Note di utilizzo per la concessione dell'autorizzazione ASSUMEROLE

Le seguenti note di utilizzo si applicano alla concessione dell'autorizzazione ASSUMEROLE in Amazon Redshift.

Utilizza l'autorizzazione ASSUMEROLE per controllare le autorizzazioni di accesso del ruolo IAM per gli utenti, i ruoli o i gruppi del database sui comandi quali COPY, UNLOAD, EXTERNAL FUNCTION o CREATE MODEL. Dopo aver concesso l'autorizzazione ASSUMEROLE a un utente, un ruolo o un gruppo per un ruolo IAM, l'utente, il ruolo o il gruppo può assumere tale ruolo all'esecuzione del comando. L'autorizzazione ASSUMEROLE consente di concedere l'accesso ai comandi appropriati in base alle esigenze.

Solo un utente del database con privilegi avanzati può concedere o revocare l'autorizzazione ASSUMEROLE per utenti, ruoli e gruppi. Un utente con privilegi avanzati mantiene sempre l'autorizzazione ASSUMEROLE.

Per abilitare l'utilizzo dell'autorizzazione ASSUMEROLE per utenti, ruoli e gruppi, un utente con privilegi avanzati esegue le due operazioni seguenti:

- Esegue l'istruzione di seguito una volta nel cluster:

```
revoke assumerole on all from public for all;
```

- Concessione dell'autorizzazione ASSUMEROLE a utenti, ruoli e gruppi per i comandi appropriati.

Quando si concede l'autorizzazione ASSUMEROLE, è possibile specificare il concatenamento dei ruoli nella clausola ON. Per separare i ruoli in una catena di ruoli è possibile utilizzare le virgole, ad esempio `Role1,Role2,Role3`. Se il concatenamento dei ruoli è stato specificato durante la concessione dell'autorizzazione ASSUMEROLE, è necessario specificare la catena di ruoli durante l'esecuzione delle operazioni concesse dall'autorizzazione ASSUMEROLE. Non è possibile specificare singoli ruoli all'interno della catena di ruoli quando si eseguono operazioni concesse dall'autorizzazione ASSUMEROLE. Ad esempio, se a un utente, un ruolo o un gruppo viene concessa la catena di ruoli `Role1,Role2,Role3`, non è possibile specificare solo `Role1` per eseguire le operazioni.

Se un utente prova a eseguire un'operazione COPY, UNLOAD, EXTERNAL FUNCTION o CREATE MODEL e non gli è stata concessa l'autorizzazione ASSUMEROLE, viene visualizzato un messaggio simile al seguente.

```
ERROR:  User awsuser does not have ASSUMEROLE permission on IAM role
 "arn:aws:iam::123456789012:role/RoleA" for COPY
```

Per elencare gli utenti a cui è stato concesso l'accesso ai ruoli IAM e ai comandi tramite l'autorizzazione ASSUMEROLE, consultare HAS_ASSUMEROLE_PRIVILEGE. Per elencare i ruoli IAM e le autorizzazioni dei comandi concesse a un utente specifico, consultare PG_GET_IAM_ROLE_BY_USER. Per elencare gli utenti, i ruoli e i gruppi a cui è stato concesso l'accesso a un ruolo IAM specifico, consulta PG_GET_GRANTEE_BY_IAM_ROLE.

Note di utilizzo per la concessione delle autorizzazioni di machine learning

Non è possibile concedere o revocare direttamente le autorizzazioni relative a una funzione ML. Una funzione ML appartiene a un modello ML e le autorizzazioni sono controllate tramite il modello. È invece possibile concedere le autorizzazioni relative al modello ML. L'esempio seguente dimostra come concedere le autorizzazioni a tutti gli utenti per eseguire la funzione ML associata al modello `customer_churn`.

```
GRANT EXECUTE ON MODEL customer_churn TO PUBLIC;
```

È anche possibile concedere tutte le autorizzazioni a un utente per il modello ML `customer_churn`.

```
GRANT ALL on MODEL customer_churn TO ml_user;
```

La concessione dell'autorizzazione EXECUTE relativa a una funzione ML avrà esito negativo se nello schema è presente una funzione ML, anche se tale funzione ML dispone già dell'autorizzazione

EXECUTEtramite GRANT EXECUTE ON MODEL. Si consiglia di utilizzare uno schema separato quando si utilizza il comando CREATE MODEL per mantenere le funzioni ML in uno schema separato. L'esempio seguente mostra come fare.

```
CREATE MODEL ml_schema.customer_churn
FROM customer_data
TARGET churn
FUNCTION ml_schema.customer_churn_prediction
IAM_ROLE default
SETTINGS (
 S3_BUCKET 'your-s3-bucket'
);
```

Esempi

L'esempio seguente concede all'utente il privilegio SELECT sulla tabella SALES fred.

```
grant select on table sales to fred;
```

L'esempio seguente concede all'utente il privilegio SELECT su tutte le tabelle dello schema QA_TICKIT fred.

```
grant select on all tables in schema qa_tickit to fred;
```

L'esempio seguente concede tutti i privilegi dello schema sullo schema QA_TICKIT al gruppo di utenti QA_USERS. I privilegi dello schema sono CREATE e USAGE. USAGE concede agli utenti l'accesso agli oggetti nello schema, ma non concede i privilegi come INSERT o SELECT su quegli oggetti. Concedere i privilegi a ciascun oggetto separatamente.

```
create group qa_users;
grant all on schema qa_tickit to group qa_users;
```

L'esempio seguente concede tutti i privilegi sulla tabella SALES nello schema QA_TICKIT a tutti gli utenti del gruppo QA_USERS.

```
grant all on table qa_tickit.sales to group qa_users;
```

L'esempio seguente concede tutti i privilegi sulla tabella SALES nello schema QA_TICKIT a tutti gli utenti del gruppo QA_USERS e RO_USERS.

```
grant all on table qa_tickit.sales to group qa_users, group ro_users;
```

L'esempio seguente concede il privilegio DROP sulla tabella SALES nello schema QA_TICKIT a tutti gli utenti del gruppo QA_USERS.

```
grant drop on table qa_tickit.sales to group qa_users;>
```

La seguente sequenza di comandi mostra come l'accesso a uno schema non concede privilegi su una tabella nello schema.

```
create user schema_user in group qa_users password 'Abcd1234';
create schema qa_tickit;
create table qa_tickit.test (col1 int);
grant all on schema qa_tickit to schema_user;

set session authorization schema_user;
select current_user;


current_user
--------------
schema_user
(1 row)


select count(*) from qa_tickit.test;


ERROR: permission denied for relation test [SQL State=42501]


set session authorization dw_user;
grant select on table qa_tickit.test to schema_user;
set session authorization schema_user;
select count(*) from qa_tickit.test;


count
-------
0
(1 row)
```

La seguente sequenza di comandi mostra come l'accesso a una vista non implica l'accesso alle tabelle sottostanti. L'utente chiamato VIEW_USER non può selezionare dalla tabella DATE sebbene a questo utente siano stati concessi tutti i privilegi su VIEW_DATE.

```
create user view_user password 'Abcd1234';
create view view_date as select * from date;
grant all on view_date to view_user;
set session authorization view_user;
select current_user;


current_user
--------------
view_user
(1 row)


select count(*) from view_date;


count
-------
365
(1 row)


select count(*) from date;


ERROR:  permission denied for relation date
```

Nell'esempio seguente vengono concessi privilegi SELECT sulle colonne `cust_name` e `cust_phone` della tabella `cust_profile` all'utente `user1`.

```
grant select(cust_name, cust_phone) on cust_profile to user1;
```

L'esempio seguente concede il privilegio SELECT sulle colonne `cust_name` e `cust_phone` e il privilegio UPDATE sulla colonna `cust_contact_preference` della tabella `cust_profile` tabella al gruppo `sales_group`.

```
grant select(cust_name, cust_phone), update(cust_contact_preference) on cust_profile to
 group sales_group;
```

Nell'esempio seguente viene illustrato l'utilizzo della parola chiave ALL per concedere i privilegi SELECT e UPDATE su tre colonne della tabella `cust_profile` al gruppo `sales_admin`.

```
grant ALL(cust_name, cust_phone,cust_contact_preference) on cust_profile to group
 sales_admin;
```

L'esempio seguente concede all'utente `user2` il privilegio SELECT sulla colonna `cust_name` della vista `cust_profile_vw`.

```
grant select(cust_name) on cust_profile_vw to user2;
```

Esempi di assegnazione dell'accesso alle unità di condivisione dati

Gli esempi di seguito riportano le autorizzazioni di utilizzo per la condivisione dei dati GRANT su un database o uno schema specifico creato da una unità di condivisione dati.

Nell'esempio seguente, un amministratore lato producer assegna l'autorizzazione USAGE dell'unità di condivisione dati `salesshare` allo spazio dei nomi specificato.

```
GRANT USAGE ON DATASHARE salesshare TO NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

Nell'esempio seguente, un amministratore lato consumer assegna l'autorizzazione USAGE di `sales_db` a `Bob`.

```
GRANT USAGE ON DATABASE sales_db TO Bob;
```

Nell'esempio seguente, un amministratore lato consumer assegna l'autorizzazione GRANT USAGE dello schema `sales_schema` al ruolo `Analyst_role`. `sales_schema` è uno schema esterno che fa riferimento a sales_db.

```
GRANT USAGE ON SCHEMA sales_schema TO ROLE Analyst_role;
```

A questo punto, `Bob` e `Analyst_role` possono accedere a tutti gli oggetti del database in `sales_schema` e `sales_db`.

L'esempio seguente mostra l'assegnazione di autorizzazioni aggiuntive per gli oggetti in un database condiviso. Queste autorizzazioni aggiuntive sono necessarie solo se il comando CREATE DATABASE utilizzato per creare il database condiviso includeva la clausola WITH PERMISSIONS. Se il comando CREATE DATABASE non includeva WITH PERMISSIONS, l'assegnazione di USAGE per il database condiviso consente l'accesso completo a tutti gli oggetti in quel database.

```
GRANT SELECT ON sales_db.sales_schema.tickit_sales_redshift to Bob;
```

Esempi di assegnazione di autorizzazioni con ambito

L'esempio seguente assegna al ruolo `Sales` l'utilizzo di tutti gli schemi attuali e futuri nel database `Sales_db`.

```
GRANT USAGE FOR SCHEMAS IN DATABASE Sales_db TO ROLE Sales;
```

L'esempio seguente assegna all'utente `alice` l'autorizzazione SELECT per tutte le tabelle correnti e future del database `Sales_db` e assegna ad `alice` l'autorizzazione a fornire ad altri utenti le autorizzazioni con ambito per le tabelle in `Sales_db`.

```
GRANT SELECT FOR TABLES IN DATABASE Sales_db TO alice WITH GRANT OPTION;
```

L'esempio seguente concede all'utente bob l'autorizzazione EXECUTE per le funzioni dello schema `Sales_schema`.

```
GRANT EXECUTE FOR FUNCTIONS IN SCHEMA Sales_schema TO bob;
```

L'esempio seguente concede al ruolo `Sales` tutte le autorizzazioni per tutte le tabelle dello schema `ShareSchema` del database `ShareDb`. Quando si specifica lo schema, è possibile indicare il database dello schema utilizzando il formato in due parti `database.schema`.

```
GRANT ALL FOR TABLES IN SCHEMA ShareDb.ShareSchema TO ROLE Sales;
```

L'esempio seguente è uguale al precedente. Puoi specificare il database utilizzando la parola chiave `DATABASE` anziché utilizzare un formato in due parti.

```
GRANT ALL FOR TABLES IN SCHEMA ShareSchema DATABASE ShareDb TO ROLE Sales;
```

Esempi di concessione del privilegio ASSUMEROLE

Di seguito sono riportati esempi di concessione del privilegio ASSUMEROLE.

Nell'esempio seguente viene illustrata l'istruzione REVOKE che un utente con privilegi avanzati esegue una volta nel cluster per abilitare l'utilizzo del privilegio ASSUMEROLE per utenti e gruppi. Quindi, l'utente con privilegi avanzati concede il privilegio ASSUMEROLE a utenti e gruppi per i comandi appropriati. Per informazioni su come abilitare l'utilizzo del privilegio ASSUMEROLE per utenti e gruppi, consultare Note di utilizzo per la concessione dell'autorizzazione ASSUMEROLE.

```
revoke assumerole on all from public for all;
```

L'esempio seguente concede il privilegio ASSUMEROLE all'utente `reg_user1` per il ruolo IAM `Redshift-S3-Read` per eseguire operazioni COPY.

```
grant assumerole on 'arn:aws:iam::123456789012:role/Redshift-S3-Read'
to reg_user1 for copy;
```

L'esempio seguente concede il privilegio ASSUMEROLE all'utente `reg_user1` per la catena di ruoli IAM `RoleA`, `RoleB` per eseguire operazioni COPY.

```
grant assumerole
on 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB'
to reg_user1
for unload;
```

Di seguito è riportato un esempio del comando UNLOAD utilizzando la catena di ruoli IAM `RoleA`, `RoleB`.

```
unload ('select * from venue limit 10')
to 's3://companyb/redshift/venue_pipe_'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

L'esempio seguente concede il privilegio ASSUMEROLE all'utente `reg_user1` per il ruolo IAM `Redshift-Exfunc` per creare funzioni esterne.

```
grant assumerole on 'arn:aws:iam::123456789012:role/Redshift-Exfunc'
to reg_user1 for external function;
```

L'esempio seguente concede il privilegio ASSUMEROLE all'utente `reg_user1` per il ruolo IAM `Redshift-model` per creare modelli di machine learning.

```
grant assumerole on 'arn:aws:iam::123456789012:role/Redshift-ML'
to reg_user1 for create model;
```

Esempi di concessione dei privilegi ROLE

L'esempio seguente concede sample_role1 a user1.

```
CREATE ROLE sample_role1;
GRANT ROLE sample_role1 TO user1;
```

L'esempio seguente concede sample_role1 a user1 con l'opzione WITH ADMIN OPTION, imposta la sessione corrente per user1 e user1 concede sample_role1 a user2.

```
GRANT ROLE sample_role1 TO user1 WITH ADMIN OPTION;
SET SESSION AUTHORIZATION user1;
GRANT ROLE sample_role1 TO user2;
```

L'esempio seguente concede sample_role1 a sample_role2.

```
GRANT ROLE sample_role1 TO ROLE sample_role2;
```

L'esempio seguente concede sample_role2 a sample_role3 e sample_role4. Quindi tenta di concedere sample_role3 a sample_role1.

```
GRANT ROLE sample_role2 TO ROLE sample_role3;
GRANT ROLE sample_role3 TO ROLE sample_role2;
ERROR: cannot grant this role, a circular dependency was detected between these roles
```

L'esempio seguente concede i privilegi di sistema CREATE USER a sample_role1.

```
GRANT CREATE USER TO ROLE sample_role1;
```

L'esempio seguente concede il ruolo definito dal sistema `sys:dba` a user1.

```
GRANT ROLE sys:dba TO user1;
```

L'esempio seguente tenta di concedere sample_role3 in una dipendenza circolare a sample_role2.

```
CREATE ROLE sample_role3;
GRANT ROLE sample_role2 TO ROLE sample_role3;
GRANT ROLE sample_role3 TO ROLE sample_role2; -- fail
ERROR:  cannot grant this role, a circular dependency was detected between these roles
```

INSERT

Argomenti

Inserisce nuove righe in una tabella. Puoi inserire una singola riga con la sintassi VALUES, più righe con la sintassi VALUES o una o più righe definite dai risultati di una query (INSERT INTO...SELECT).

> **Note**
>
> Consigliamo l'uso del comando COPY per caricare grandi quantità di dati. L'utilizzo di singole istruzioni INSERT per popolare una tabella potrebbe essere eccessivamente lento. In alternativa, se i dati esistono già in altre tabelle di database Amazon Redshift, utilizzare INSERT INTO SELECT o CREATE TABLE AS per migliorare le prestazioni. Per ulteriori informazioni sull'utilizzo del comando COPY per caricare le tabelle, vedi Caricamento dei dati.

> **Note**
>
> Le dimensioni massime per una istruzione SQL è di 16 MB.

Sintassi

```
INSERT INTO table_name [ ( column [, ...] ) ]
{DEFAULT VALUES |
VALUES ( { expression | DEFAULT } [, ...] )
```

```
[, ( { expression | DEFAULT } [, ...] )
[, ...] ] |
query }
```

Parametri

table_name

Tabella temporanea o persistente. Solo il proprietario della tabella o un utente con il privilegio INSERT sulla tabella può inserire le righe. Se utilizzi la clausola query per inserire le righe, devi avere il privilegio SELECT sulle tabelle denominate nella query.

> **Note**
>
> Utilizza INSERT (tabella esterna) per inserire i risultati di una query SELECT nelle tabelle esistenti nel catalogo esterno. Per ulteriori informazioni, consulta INSERT (tabella esterna).

column

Puoi inserire i valori in una o più colonne della tabella. Puoi elencare i nomi delle colonne di destinazione in qualsiasi ordine. Se non specifichi un elenco di colonne, i valori da inserire devono corrispondere alle colonne della tabella nell'ordine in cui sono stati dichiarati nell'istruzione CREATE TABLE. Se il numero di valori da inserire è inferiore al numero di colonne nella tabella, vengono caricate le prime n colonne.

Il valore predefinito dichiarato o un valore nullo viene caricato in ogni colonna non elencata (implicitamente o esplicitamente) nell'istruzione INSERT.

DEFAULT VALUES

Se alle colonne della tabella sono stati assegnati i valori predefiniti al momento della creazione della tabella, utilizza queste parole chiave per inserire una riga costituita interamente da valori predefiniti. Se le colonne non hanno valori predefiniti, in quelle colonne vengono inseriti i valori null. Se una delle colonne è dichiarata NOT NULL, l'istruzione INSERT restituisce un errore.

VALUES

Utilizza questa parola chiave per inserire una o più righe, con ogni riga composta da uno o più valori. L'elenco VALUES per ogni riga deve essere allineato all'elenco delle colonne. Per inserire

più righe, utilizza una virgola come delimitatore tra ciascun elenco di espressioni. Non ripetere la parola chiave VALUES. Tutti gli elenchi VALUES per un'istruzione INSERT a più righe devono contenere lo stesso numero di valori.

espressione

Un singolo valore o un'espressione che valuta un singolo valore. Ogni valore deve essere compatibile con il tipo di dati della colonna in cui viene inserito. Se possibile, un valore il cui tipo di dati non corrisponde al tipo di dati dichiarato della colonna viene automaticamente convertito in un tipo di dati compatibile. Ad esempio:

- Un valore decimale `1.1` è inserito in una colonna INT come `1`.
- Un valore decimale `100.8976` è inserito in una colonna DEC(5,2) come `100.90`.

Puoi convertire esplicitamente un valore in un tipo di dati compatibile includendo la sintassi del cast di tipo nell'espressione. Ad esempio, se la colonna COL1 nella tabella T1 è una colonna CHAR (3):

```
insert into t1(col1) values('Incomplete'::char(3));
```

L'istruzione inserisce il valore `Inc` nella colonna.

Per un'istruzione INSERT VALUES a riga singola, puoi utilizzare una sottoquery scalare come espressione. Il risultato della sottoquery viene inserito nella colonna appropriata.

> **Note**
>
> Le sottoquery non sono supportate come espressioni per le istruzioni INSERT VALUES a più righe.

DEFAULT

Usa questa parola chiave per inserire il valore predefinito per una colonna, come definito al momento della creazione della tabella. Se non esiste un valore predefinito per una colonna, viene inserito un valore nullo. Non puoi inserire un valore predefinito in una colonna con un vincolo NOT NULL se tale colonna non ha un valore predefinito esplicito assegnato nell'istruzione CREATE TABLE.

query

Inserisci una o più righe nella tabella definendo qualsiasi query. Tutte le righe prodotte dalla query vengono inserite nella tabella. La query deve restituire un elenco di colonne compatibile con le colonne della tabella, ma i nomi delle colonne non devono corrispondere.

Note per l'utilizzo

> **Note**
>
> Consigliamo l'uso del comando COPY per caricare grandi quantità di dati. L'utilizzo di singole istruzioni INSERT per popolare una tabella potrebbe essere eccessivamente lento. In alternativa, se i dati esistono già in altre tabelle di database Amazon Redshift, utilizzare INSERT INTO SELECT o CREATE TABLE AS per migliorare le prestazioni. Per ulteriori informazioni sull'utilizzo del comando COPY per caricare le tabelle, vedi Caricamento dei dati.

Il formato dei dati per i valori inseriti deve corrispondere al formato dei dati specificato dalla definizione CREATE TABLE.

Dopo aver inserito un numero elevato di nuove righe in una tabella:

- Applica l'operazione di vacuum alla tabella per recuperare spazio di memorizzazione e riordinare le righe.
- Analizza la tabella per aggiornare le statistiche del pianificatore di query.

Quando i valori vengono inseriti nelle colonne DECIMAL e superano la scala specificata, i valori caricati vengono arrotondati all'occorrenza. Per esempio, se un valore di `20.259` viene inserito in una colonna DECIMAL(8,2) il valore che viene memorizzato è `20.26`.

Puoi inserire in una colonna GENERATED BY DEFAULT AS IDENTITY. Puoi aggiornare colonne definite come GENERATED BY DEFAULT AS IDENTITY con i valori forniti. Per ulteriori informazioni, consulta GENERATED BY DEFAULT AS IDENTITY.

Esempi di INSERT

La tabella CATEGORY nel database TICKIT contiene le seguenti righe:

```
catid | catgroup |  catname  |                catdesc
```

```
-------+----------+-----------+--------------------------------------------
     1 | Sports   | MLB       | Major League Baseball
     2 | Sports   | NHL       | National Hockey League
     3 | Sports   | NFL       | National Football League
     4 | Sports   | NBA       | National Basketball Association
     5 | Sports   | MLS       | Major League Soccer
     6 | Shows    | Musicals  | Musical theatre
     7 | Shows    | Plays     | All non-musical theatre
     8 | Shows    | Opera     | All opera and light opera
     9 | Concerts | Pop       | All rock and pop music concerts
    10 | Concerts | Jazz      | All jazz singers and bands
    11 | Concerts | Classical | All symphony, concerto, and choir concerts
(11 rows)
```

Crea una tabella CATEGORY_STAGE con uno schema simile alla tabella CATEGORY, ma definisci i valori predefiniti per le colonne:

```
create table category_stage
(catid smallint default 0,
catgroup varchar(10) default 'General',
catname varchar(10) default 'General',
catdesc varchar(50) default 'General');
```

La seguente istruzione INSERT seleziona tutte le righe dalla tabella CATEGORY e le inserisce nella tabella CATEGORY_STAGE.

```
insert into category_stage
(select * from category);
```

Le parentesi che racchiudono la query sono facoltative.

Questo comando inserisce una nuova riga nella tabella CATEGORY_STAGE con un valore specificato per ogni colonna nell'ordine:

```
insert into category_stage values
(12, 'Concerts', 'Comedy', 'All stand-up comedy performances');
```

Puoi anche inserire una nuova riga che combina valori specifici e valori predefiniti:

```
insert into category_stage values
```

```
(13, 'Concerts', 'Other', default);
```

Esegui la seguente query per restituire le righe inserite:

```
select * from category_stage
where catid in(12,13) order by 1;

 catid | catgroup | catname |             catdesc
-------+----------+---------+----------------------------------
    12 | Concerts | Comedy  | All stand-up comedy performances
    13 | Concerts | Other   | General
(2 rows)
```

Gli esempi seguenti mostrano alcune istruzioni INSERT VALUES a più righe. Il primo esempio inserisce valori CATID specifici per due righe e valori predefiniti per le altre colonne in entrambe le righe.

```
insert into category_stage values
(14, default, default, default),
(15, default, default, default);

select * from category_stage where catid in(14,15) order by 1;
 catid | catgroup | catname | catdesc
-------+----------+---------+---------
    14 | General  | General | General
    15 | General  | General | General
(2 rows)
```

Il prossimo esempio inserisce tre righe con varie combinazioni di valori specifici e predefiniti:

```
insert into category_stage values
(default, default, default, default),
(20, default, 'Country', default),
(21, 'Concerts', 'Rock', default);

select * from category_stage where catid in(0,20,21) order by 1;
 catid | catgroup | catname | catdesc
-------+----------+---------+---------
     0 | General  | General | General
    20 | General  | Country | General
    21 | Concerts | Rock    | General
```

```
(3 rows)
```

La prima serie di VALUES in questo esempio produce gli stessi risultati della specifica di DEFAULT VALUES per un'istruzione INSERT a riga singola.

I seguenti esempi mostrano il comportamento di INSERT quando una tabella ha una colonna IDENTITY. Innanzitutto, crea una nuova versione della tabella CATEGORY, quindi inserisci le righe in essa contenute da CATEGORY:

```
create table category_ident
(catid int identity not null,
catgroup varchar(10) default 'General',
catname varchar(10) default 'General',
catdesc varchar(50) default 'General');


insert into category_ident(catgroup,catname,catdesc)
select catgroup,catname,catdesc from category;
```

Non è possibile inserire valori interi specifici nella colonna CATID IDENTITY. I valori delle colonne IDENTITY vengono generati automaticamente.

L'esempio seguente dimostra che le sottoquery non possono essere utilizzate come espressioni in istruzioni INSERT VALUES a più righe:

```
insert into category(catid) values
((select max(catid)+1 from category)),
((select max(catid)+2 from category));

ERROR: can't use subqueries in multi-row VALUES
```

L'esempio seguente mostra un inserimento in una tabella temporanea popolata con i dati della tabella `venue` utilizzando la clausola `WITH SELECT`. Per ulteriori informazioni sulla tabella `venue`, consulta Database di esempio.

Per prima cosa, crea la tabella temporanea `#venuetemp`.

```
CREATE TABLE #venuetemp AS SELECT * FROM venue;
```

Elenca le righe nella tabella `#venuetemp`.

```
SELECT * FROM #venuetemp ORDER BY venueid;

venueid | venuename                | venuecity  | venuestate| venueseats
--------+--------------------------+------------+-----------+------------
1       Toyota Park                Bridgeview   IL          0
2       Columbus Crew Stadium      Columbus     OH          0
3       RFK Stadium                Washington   DC          0
4       CommunityAmerica Ballpark  Kansas City  KS          0
5       Gillette Stadium           Foxborough   MA          68756
...
```

Inserisci 10 righe duplicate nella tabella #venuetemp utilizzando la clausola WITH SELECT.

```
INSERT INTO #venuetemp (WITH venuecopy AS (SELECT * FROM venue) SELECT * FROM venuecopy
 ORDER BY 1 LIMIT 10);
```

Elenca le righe nella tabella #venuetemp.

```
SELECT * FROM #venuetemp ORDER BY venueid;

venueid | venuename                | venuecity  | venuestate| venueseats
--------+--------------------------+------------+-----------+------------
1       Toyota Park                Bridgeview   IL          0
1       Toyota Park                Bridgeview   IL          0
2       Columbus Crew Stadium      Columbus     OH          0
2       Columbus Crew Stadium      Columbus     OH          0
3       RFK Stadium                Washington   DC          0
3       RFK Stadium                Washington   DC          0
4       CommunityAmerica Ballpark  Kansas City  KS          0
4       CommunityAmerica Ballpark  Kansas City  KS          0
5       Gillette Stadium           Foxborough   MA          68756
5       Gillette Stadium           Foxborough   MA          68756
...
```

INSERT (tabella esterna)

Inserisce i risultati di una query SELECT in tabelle esterne esistenti su un catalogo esterno AWS Glue AWS Lake Formation, ad esempio for o un metastore Apache Hive. Utilizza lo stesso ruolo AWS Identity and Access Management (IAM) utilizzato per il comando CREATE EXTERNAL SCHEMA per interagire con cataloghi esterni e Amazon S3.

Per le tabelle non partizionate, il comando INSERT (tabella esterna) scrive i dati nella posizione Amazon S3 definita nella tabella, in base alle proprietà della tabella e al formato di file specificati.

Per le tabelle con partizioni, INSERT (tabella esterna) scrive i dati nella posizione Amazon S3 in base alla chiave di partizione specificata nella tabella. Al termine dell'operazione INSERT, inoltre, registra automaticamente le nuove partizioni nel catalogo esterno.

Non è possibile eseguire INSERT (tabella esterna) all'interno di un blocco di transazione (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

Sintassi

```
INSERT INTO external_schema.table_name
{ select_statement }
```

Parametri

external_schema.table_name

Il nome di uno schema esterno e di una tabella esterna di destinazione esistenti in cui effettuare l'inserimento.

select_statement

Istruzione che inserisce una o più righe nella tabella esterna definendo qualsiasi query. Tutte le righe prodotte dalla query vengono scritte in Amazon S3 in formato testo o Parquet in base alla definizione della tabella. La query deve restituire un elenco di colonne compatibile con i tipi di dati delle colonne nella tabella esterna. Tuttavia, i nomi delle colonne non devono necessariamente corrispondere.

Note per l'utilizzo

Il numero di colonne nella query SELECT deve essere uguale alla somma delle colonne di dati e delle colonne di partizione. La posizione e il tipo di dati di ogni colonna di dati devono corrispondere a quelli della tabella esterna. La posizione delle colonne di partizione deve essere alla fine della query SELECT, nello stesso ordine in cui sono state definite nel comando CREATE EXTERNAL TABLE. I nomi delle colonne non devono necessariamente corrispondere.

In alcuni casi, è possibile eseguire il comando INSERT (tabella esterna) su un catalogo dati AWS Glue o un metastore Hive. Nel caso di AWS Glue, il ruolo IAM utilizzato per creare lo schema esterno

deve disporre di autorizzazioni di lettura e scrittura su Amazon AWS Glue S3 e. Se utilizzi un AWS Lake Formation catalogo, questo ruolo IAM diventa il proprietario della nuova tabella Lake Formation. Questo ruolo IAM deve disporre almeno delle seguenti autorizzazioni:

- Autorizzazione SELECT, INSERT, UPDATE sulla tabella esterna
- Autorizzazione per la posizione dati sul percorso Amazon S3 della tabella esterna

Per garantire che i nomi file siano univoci, Amazon Redshift utilizza il seguente formato per il nome di ogni file caricato in Amazon S3 per impostazione predefinita.

*<date>*_*<time>*_*<microseconds>*_*<query_id>*_*<slice-number>*_part_*<part-number>*.*<format>*.

Un esempio è `20200303_004509_810669_1007_0001_part_00.parquet`.

Quando si esegue il comando INSERT (tabella esterna), considerare quanto segue:

- Le tabelle esterne con un formato diverso da PARQUET o TEXTFILE non sono supportate.
- Questo comando supporta le proprietà esistenti della tabella come 'write.parallel', 'write.maxfilesize.mb', 'compression_type' e 'serialization.null.format'. Per aggiornare tali valori, eseguire il comando ALTER TABLE SET TABLE PROPERTIES.
- La proprietà della tabella 'numRows' viene aggiornata automaticamente verso la fine dell'operazione INSERT. La proprietà della tabella deve essere definita o aggiunta alla tabella se già non è stata creata dall'operazione CREATE EXTERNAL TABLE AS.
- La clausola LIMIT non è supportata nella query SELECT esterna. Usa invece una clausola LIMIT nidificata.
- È possibile utilizzare la tabella STL_UNLOAD_LOG per tenere traccia dei file scritti su Amazon S3 da ogni operazione INSERT (tabella esterna).
- Amazon Redshift supporta solo la crittografia standard Amazon S3 per INSERT (tabella esterna).

Esempi di INSERT (tabella esterna)

Nell'esempio seguente i risultati dell'istruzione SELECT vengono inseriti nella tabella esterna.

```
INSERT INTO spectrum.lineitem
SELECT * FROM local_lineitem;
```

Nell'esempio seguente i risultati dell'istruzione SELECT vengono inseriti in una tabella esterna partizionata mediante il partizionamento statico. Le colonne di partizione nell'istruzione SELECT sono hardcoded. Le colonne delle partizioni devono trovarsi alla fine della query.

```
INSERT INTO spectrum.customer
SELECT name, age, gender, 'May', 28 FROM local_customer;
```

Nell'esempio seguente i risultati dell'istruzione SELECT vengono inseriti in una tabella esterna partizionata utilizzando il partizionamento dinamico. Le colonne delle partizioni non sono hardcoded. I dati vengono aggiunti automaticamente alle cartelle delle partizioni esistenti o alle nuove cartelle se viene aggiunta una nuova partizione.

```
INSERT INTO spectrum.customer
SELECT name, age, gender, month, day FROM local_customer;
```

LOCK

Limita l'accesso a una tabella del database. Questo comando è significativo solo quando viene eseguito all'interno di un blocco di transazione.

Il comando LOCK ottiene un blocco a livello di tabella in modalità "ACCESS EXCLUSIVE", in attesa, se necessario, di eventuali blocchi in conflitto da rilasciare. Bloccare esplicitamente una tabella in questo modo fa sì che le letture e le scritture sulla tabella restino in attesa quando vengono tentate da altre transazioni o sessioni. Un blocco esplicito della tabella creato da un utente impedisce temporaneamente a un altro utente di selezionare i dati da quella tabella o di caricare dati in essa. Il blocco viene rilasciato quando la transazione che contiene il comando LOCK viene completata.

I blocchi di tabella meno restrittivi vengono acquisiti implicitamente da comandi che fanno riferimento a tabelle, ad esempio operazioni di scrittura. Ad esempio, se un utente tenta di leggere i dati da una tabella mentre un altro utente sta aggiornando la tabella, i dati letti saranno uno snapshot dei dati che sono già stati sottoposti al commit. (In alcuni casi, le query vengono interrotte in modo anomalo se violano le regole di isolamento serializzabili.) Per informazioni, consultare Gestione delle operazioni di scrittura simultanee.

Alcune operazioni DDL, come DROP TABLE e TRUNCATE, creano blocchi esclusivi. Queste operazioni impediscono le letture dei dati.

Se si verifica un conflitto di blocco, Amazon Redshift visualizza un messaggio di errore per avvisare l'utente che ha avviato la transazione in conflitto. La transazione che ha ricevuto il conflitto di blocco

viene interrotta. Ogni volta che si verifica un conflitto di blocco, Amazon Redshift scrive una voce nella tabella STL_TR_CONFLICT.

Sintassi

```
LOCK [ TABLE ] table_name [, ...]
```

Parametri

TABLE

Parola chiave facoltativa.

table_name

Nome della tabella da bloccare. Puoi bloccare più di una tabella utilizzando un elenco di nomi di tabella delimitati da virgole. Non è possibile bloccare le viste.

Esempio

```
begin;

lock event, sales;

...
```

MERGE

Unisce in modo condizionale le righe da una tabella di origine a una tabella di destinazione. In genere, ciò può essere ottenuto solo utilizzando separatamente più istruzioni di inserimento, aggiornamento o eliminazione. Per ulteriori informazioni sulle operazioni che MERGE consente di combinare, consulta UPDATE, DELETE e INSERT.

Sintassi

```
MERGE INTO target_table
USING source_table [ [ AS ] alias ]
ON match_condition
[ WHEN MATCHED THEN { UPDATE SET col_name = { expr } [,...] | DELETE }
WHEN NOT MATCHED THEN INSERT [ ( col_name [,...] ) ] VALUES ( { expr } [, ...] ) |
```

```
REMOVE DUPLICATES ]
```

Parametri

target_table

La tabella temporanea o permanente in cui si include l'istruzione MERGE.

source_table

La tabella temporanea o permanente che fornisce le righe da unire in target_table. source_table può essere anche una tabella Spectrum. source_table non può essere una vista o una sottoquery.

alias

Nome alternativo temporaneo per source_table.

Questo parametro è facoltativo. Anche il precedente alias con AS è facoltativo.

match_condition

Specifica predicati uguali tra la colonna della tabella di origine e la colonna della tabella di destinazione che vengono utilizzati per determinare se le righe in source_table possono essere abbinate alle righe in target_table. Se la condizione è soddisfatta, MERGE esegue matched_clause per quella riga. In caso contrario, MERGE esegue not_matched_clause per quella riga.

WHEN MATCHED

Specifica l'operazione da eseguire quando la condizione di corrispondenza tra una riga di origine e una riga di destinazione risulta True. È possibile specificare un'azione UPDATE o un'azione DELETE.

UPDATE

Aggiorna la riga corrispondente in target_table. Vengono aggiornati solo i valori nel col_name specificato.

DELETE

Elimina la riga corrispondente in target_table.

WHEN NOT MATCHED

Specifica l'operazione da eseguire quando la condizione di corrispondenza risulta False o Unknown. È possibile specificare solo l'azione di inserimento INSERT per questa clausola.

INSERT

Inserisce una riga in target_table. Il col_name di destinazione può essere elencato in qualsiasi ordine. Se non viene fornito alcun valore col_name, per impostazione predefinita l'ordine è quello di tutte le colonne della tabella nel loro ordine dichiarato.

nome_col

Una o più nomi di colonne da modificare. Non includere il nome della tabella quando specifichi la colonna di destinazione.

expr

L'espressione che definisce il nuovo valore per col_name.

REMOVE DUPLICATES

Specifica che il comando MERGE viene eseguito in modalità semplificata. La modalità semplificata presenta i seguenti requisiti:

- target_table e source_table devono avere lo stesso numero di colonne e tipi di colonna compatibili.
- Ometti la clausola WHEN e le clausole UPDATE e INSERT dal comando MERGE.
- Usa la clausola REMOVE DUPLICATES nel comando MERGE.

In modalità semplificata, MERGE esegue le seguenti operazioni:

- Le righe in target_table che hanno una corrispondenza in source_table vengono aggiornate in modo che corrispondano ai valori in source_table.
- Le righe in source_table che non hanno una corrispondenza in target_table vengono inserite in target_table.
- Quando più righe in target_table corrispondono alla stessa riga in source_table, le righe duplicate vengono rimosse. Amazon Redshift mantiene una riga e la aggiorna. Le righe duplicate che non corrispondono a una riga in source_table restano invariate.

L'utilizzo di REMOVE DUPLICATES offre prestazioni migliori rispetto all'utilizzo di WHEN MATCHED e WHEN NOT MATCHED. Consigliamo di utilizzare REMOVE DUPLICATES se target_table e source_table sono compatibili e non è necessario conservare le righe duplicate in target_table.

Note per l'utilizzo

- Per eseguire le istruzioni MERGE, devi essere il proprietario sia di source_table che di target_table o disporre dell'autorizzazione SELECT per tali tabelle. Inoltre, devi disporre delle autorizzazioni UPDATE, DELETE e INSERT per target_table, a seconda delle operazioni incluse nell'istruzione MERGE.
- target_table non può essere una tabella di sistema, una tabella di catalogo o una tabella esterna.
- source_table e target_table non possono essere la stessa tabella.
- Non è possibile utilizzare la clausola WITH in un'istruzione MERGE.
- Le righe in target_table non possono corrispondere a più righe in source_table.

  Considera il seguente esempio:

```
CREATE TABLE target (id INT, name CHAR(10));
CREATE TABLE source (id INT, name CHAR(10));

INSERT INTO target VALUES (1, 'Bob'), (2, 'John');
INSERT INTO source VALUES (1, 'Tony'), (1, 'Alice'), (3, 'Bill');

MERGE INTO target USING source ON target.id = source.id
WHEN MATCHED THEN UPDATE SET id = source.id, name = source.name
WHEN NOT MATCHED THEN INSERT VALUES (source.id, source.name);
ERROR: Found multiple matches to update the same tuple.

MERGE INTO target USING source ON target.id = source.id
WHEN MATCHED THEN DELETE
WHEN NOT MATCHED THEN INSERT VALUES (source.id, source.name);
ERROR: Found multiple matches to update the same tuple.
```

  In entrambe le istruzioni MERGE, l'operazione non riesce perché nella tabella `source` sono presenti più righe con un valore ID di 1.
- match_condition ed expr non possono fare riferimento parzialmente a colonne di tipo SUPER. Ad esempio, se l'oggetto di tipo SUPER è un array o una struttura, non puoi usare singoli elementi di quella colonna per match_condition o expr, ma puoi utilizzare l'intera colonna.

  Considera il seguente esempio:

```
CREATE TABLE IF NOT EXISTS target (key INT, value SUPER);
CREATE TABLE IF NOT EXISTS source (key INT, value SUPER);
```

```
INSERT INTO target VALUES (1, JSON_PARSE('{"key": 88}'));
INSERT INTO source VALUES (1, ARRAY(1, 'John')), (2, ARRAY(2, 'Bill'));

MERGE INTO target USING source ON target.key = source.key
WHEN matched THEN UPDATE SET value = source.value[0]
WHEN NOT matched THEN INSERT VALUES (source.key, source.value[0]);
ERROR: Partial reference of SUPER column is not supported in MERGE statement.
```

Per ulteriori informazioni sul tipo SUPER, consulta Tipo SUPER.

- Se source_table è di grandi dimensioni, la definizione delle colonne di join sia da target_table che da source_table come chiavi di distribuzione può migliorare le prestazioni.
- Per utilizzare la clausola REMOVE DUPLICATES, sono necessarie le autorizzazioni SELECT, INSERT e DELETE per target_table.

Esempi

L'esempio seguente crea due tabelle, quindi esegue un'operazione MERGE su di esse, aggiornando le righe corrispondenti nella tabella di destinazione e inserendo righe che prive di corrispondenze. Quindi inserisce un altro valore nella tabella di origine ed esegue un'altra operazione MERGE, questa volta eliminando le righe corrispondenti e inserendo la nuova riga dalla tabella di origine.

Per prima cosa crea e compila le tabelle di origine e destinazione.

```
CREATE TABLE target (id INT, name CHAR(10));
CREATE TABLE source (id INT, name CHAR(10));

INSERT INTO target VALUES (101, 'Bob'), (102, 'John'), (103, 'Susan');
INSERT INTO source VALUES (102, 'Tony'), (103, 'Alice'), (104, 'Bill');

SELECT * FROM target;
 id  |    name
-----+------------
 101 | Bob
 102 | John
 103 | Susan
(3 rows)

SELECT * FROM source;
 id  |    name
-----+------------
```

```
 102 | Tony
 103 | Alice
 104 | Bill
(3 rows)
```

Quindi, unisci la tabella di origine con la tabella di destinazione, aggiornando la tabella di destinazione con le righe corrispondenti e inserisci le righe della tabella di origine che non hanno alcuna corrispondenza.

```
MERGE INTO target USING source ON target.id = source.id
WHEN MATCHED THEN UPDATE SET id = source.id, name = source.name
WHEN NOT MATCHED THEN INSERT VALUES (source.id, source.name);

SELECT * FROM target;
 id  |    name
-----+------------
 101 | Bob
 102 | Tony
 103 | Alice
 104 | Bill
(4 rows)
```

Tieni presente che le righe con valori ID di 102 e 103 vengono aggiornate in modo che corrispondano ai valori dei nomi della tabella di destinazione. Inoltre, nella tabella di destinazione viene inserita una nuova riga con un valore ID di 104 e il valore del nome Bill.

Quindi, inserisci una nuova riga nella tabella di origine.

```
INSERT INTO source VALUES (105, 'David');

SELECT * FROM source;
 id  |    name
-----+------------
 102 | Tony
 103 | Alice
 104 | Bill
 105 | David
(4 rows)
```

Infine, esegui un'operazione di unione eliminando le righe corrispondenti nella tabella di destinazione e inserendo le righe prive di corrispondenze.

```
MERGE INTO target USING source ON target.id = source.id
WHEN MATCHED THEN DELETE
WHEN NOT MATCHED THEN INSERT VALUES (source.id, source.name);

SELECT * FROM target;
 id  |    name
-----+------------
 101 | Bob
 105 | David
(2 rows)
```

Le righe con valori ID di 102, 103 e 104 vengono eliminate dalla tabella di destinazione e una nuova riga con un valore ID di 105 e il valore del nome David viene inserita nella tabella di destinazione.

L'esempio seguente mostra un comando MERGE che utilizza la clausola REMOVE DUPLICATES.

```
CREATE TABLE target (id INT, name CHAR(10));
CREATE TABLE source (id INT, name CHAR(10));

INSERT INTO target VALUES (30, 'Tony'), (11, 'Alice'), (23, 'Bill');
INSERT INTO source VALUES (23, 'David'), (22, 'Clarence');

MERGE INTO target USING source ON target.id = source.id REMOVE DUPLICATES;

SELECT * FROM target;
id | name
---+------------
30 | Tony
11 | Alice
23 | David
22 | Clarence
(4 rows)
```

L'esempio seguente mostra un comando MERGE che utilizza la clausola REMOVE DUPLICATES, che rimuove le righe duplicate da target_table se ci sono righe corrispondenti in source_table.

```
CREATE TABLE target (id INT, name CHAR(10));
CREATE TABLE source (id INT, name CHAR(10));

INSERT INTO target VALUES (30, 'Tony'), (30, 'Daisy'), (11, 'Alice'), (23, 'Bill'),
 (23, 'Nikki');
INSERT INTO source VALUES (23, 'David'), (22, 'Clarence');
```

```
MERGE INTO target USING source ON target.id = source.id REMOVE DUPLICATES;

SELECT * FROM target;
id | name
---+------------
30 | Tony
30 | Daisy
11 | Alice
23 | David
22 | Clarence
(5 rows)
```

Dopo l'esecuzione di MERGE, c'è solo una riga con un valore ID di 23 in target_table. Poiché non c'era alcuna riga in source_table con il valore ID 30, le due righe duplicate con valori ID 30 rimangono in target_table.

consultare anche

INSERT, UPDATE, DELETE

PREPARE

Prepara un'istruzione per l'esecuzione.

PREPARE crea un'istruzione preparata. Quando viene eseguita l'istruzione PREPARE, l'istruzione specificata (SELECT, INSERT, UPDATE o DELETE) viene analizzata, riscritta e pianificata. Quando viene emesso un comando EXECUTE per l'istruzione preparata, Amazon Redshift può facoltativamente rivedere il piano di esecuzione della query (per migliorare le prestazioni in base ai valori dei parametri specificati) prima di eseguire l'istruzione preparata.

Sintassi

```
PREPARE plan_name [ (datatype [, ...] ) ] AS statement
```

Parametri

plan_name

Nome arbitrario assegnato a questa particolare istruzione preparata. Deve essere univoco all'interno di una singola sessione e successivamente viene utilizzato per eseguire o deallocare un'istruzione precedentemente preparata.

datatype

Il tipo di dati di un parametro dell'istruzione preparata. Per fare riferimento ai parametri nella stessa istruzione preparata, utilizza $1, $2 e così via.

statement

Qualsiasi istruzione SELECT, INSERT, UPDATE o DELETE.

Note per l'utilizzo

Le istruzioni preparate possono assumere parametri: valori che vengono sostituiti nell'istruzione quando viene eseguita. Per includere i parametri in un'istruzione preparata, fornisci un elenco di tipi di dati nell'istruzione PREPARE e, nell'istruzione da preparare, fai riferimento ai parametri per posizione utilizzando la notazione $1, $2, ... Durante l'esecuzione di un'istruzione, specifica i valori effettivi per tali parametri nell'istruzione EXECUTE. Per ulteriori dettagli, consultare EXECUTE.

Le istruzioni preparate durano solo per la sessione corrente. Al termine della sessione, l'istruzione preparata viene scartata, quindi deve essere ricreata prima di essere utilizzata di nuovo. Ciò significa anche che una singola istruzione preparata non può essere utilizzata da più client di database simultanei; tuttavia, ogni client può creare la propria istruzione preparata da utilizzare. L'istruzione preparata può essere rimossa manualmente utilizzando il comando DEALLOCATE.

Le istruzioni preparate offrono il massimo vantaggio in termini di prestazioni quando una singola sessione viene utilizzata per eseguire un numero elevato di istruzioni simili. Come illustrato, per ogni nuova esecuzione di un'istruzione preparata, Amazon Redshift può rivedere nuovamente il piano di esecuzione della query per migliorare le prestazioni in base ai valori dei parametri specificati. Per esaminare il piano di esecuzione della query che Amazon Redshift ha scelto per una specifica istruzione EXECUTE, utilizzare il comando EXPLAIN.

Per ulteriori informazioni sulla pianificazione delle query e sulle statistiche raccolte da Amazon Redshift per l'ottimizzazione delle query, consultare il comando ANALYZE.

Esempi

Creare una tabella temporanea, preparare l'istruzione INSERT e quindi eseguirla:

```
DROP TABLE IF EXISTS prep1;
CREATE TABLE prep1 (c1 int, c2 char(20));
PREPARE prep_insert_plan (int, char)
AS insert into prep1 values ($1, $2);
EXECUTE prep_insert_plan (1, 'one');
EXECUTE prep_insert_plan (2, 'two');
EXECUTE prep_insert_plan (3, 'three');
DEALLOCATE prep_insert_plan;
```

Preparare un'istruzione SELECT e quindi eseguirla:

```
PREPARE prep_select_plan (int)
AS select * from prep1 where c1 = $1;
EXECUTE prep_select_plan (2);
EXECUTE prep_select_plan (3);
DEALLOCATE prep_select_plan;
```

consultare anche

DEALLOCATE, EXECUTE

REFRESH MATERIALIZED VIEW

Aggiorna una vista materializzata

Quando crei una vista materializzata, il suo contenuto riflette lo stato della tabella o delle tabelle del database sottostante in quel momento. I dati nella vista materializzata rimangono invariati, anche quando le applicazioni apportano modifiche ai dati nelle tabelle sottostanti. Per aggiornare i dati in una vista materializzata, è possibile usare l'istruzione `REFRESH MATERIALIZED VIEW` in qualsiasi momento. Eseguendo questa istruzione, Amazon Redshift identifica le modifiche apportate nella tabella o nelle tabelle di base e quindi applica tali modifiche alla vista materializzata.

Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

Sintassi

```
REFRESH MATERIALIZED VIEW mv_name
```

Parametri

mv_name

Il nome della vista materializzata da aggiornare.

Note per l'utilizzo

Solo il proprietario di una vista materializzata può eseguire un'operazione `REFRESH MATERIALIZED VIEW` su tale vista materializzata. Inoltre, il proprietario deve disporre del privilegio SELECT sulle tabelle di base sottostanti per eseguire correttamente il comando `REFRESH MATERIALIZED VIEW`.

Il comando `REFRESH MATERIALIZED VIEW` viene eseguito come transazione propria. La semantica delle transazioni Amazon Redshift viene seguita per determinare quali dati dalle tabelle di base sono visibili al comando `REFRESH` o quando le modifiche apportate dal comando `REFRESH` sono rese visibili ad altre transazioni in esecuzione in Amazon Redshift.

- Nel caso di viste materializzate incrementali, `REFRESH MATERIALIZED VIEW` utilizza solo le righe delle tabelle di base per le quali è già stato eseguito il commit. Pertanto, se l'operazione di aggiornamento viene eseguita dopo un'istruzione DML (Data Manipolation Language) nella stessa transazione, le modifiche di tale istruzione DML non sono visibili per l'aggiornamento.
- Per un aggiornamento completo di una vista materializzata, `REFRESH MATERIALIZED VIEW` vede tutte le righe della tabella di base visibili alla transazione di aggiornamento, in base alla semantica delle transazioni Amazon Redshift usuale.
- A seconda del tipo di argomento di input, Amazon Redshift supporta ancora l'aggiornamento incrementale per le viste materializzate per le seguenti funzioni con tipi di argomenti di input specifici: DATE (timestamp), DATE_PART (data, ora, intervallo, time-tz), DATE_TRUNC (timestamp, intervallo).
- L'aggiornamento incrementale è supportato in una vista materializzata in cui la tabella di base si trova in un'unità di condivisione dati.

Alcune operazioni in Amazon Redshift interagiscono con le viste materializzate. Alcune di queste operazioni potrebbero forzare un'operazione `REFRESH MATERIALIZED VIEW` per ricalcolare

completamente la vista materializzata, anche se la query che definisce la vista materializzata utilizza solo le funzionalità SQL idonee per l'aggiornamento incrementale. Ad esempio:

- Le operazioni vacuum in background potrebbero essere bloccate se le viste materializzate non vengono aggiornate. Dopo un periodo di soglia definito internamente, è possibile eseguire l'operazione di vacuum. Con l'operazione vacuum, tutte le viste materializzate dipendenti vengono contrassegnate come da rielaborare al successivo aggiornamento (anche se sono di tipo incrementale). Per informazioni su VACUUM, vedere VACUUM. Per ulteriori informazioni sugli eventi e le modifiche allo stato, consultare STL_MV_STATE.
- Alcune operazioni avviate dall'utente sulle tabelle di base impongono che una vista materializzata venga rielaborata completamente alla successiva esecuzione dell'operazione REFRESH. Esempi di tali operazioni sono un VACUUM richiamato manualmente, un ridimensionamento classico, un'operazione ALTER DISTKEY, un'operazione ALTER SORTKEY e un'operazione di troncatura. Per ulteriori informazioni sugli eventi e le modifiche allo stato, consultare STL_MV_STATE.

Aggiornamento incrementale per le viste materializzate in un datashare

Amazon Redshift supporta l'aggiornamento automatico e incrementale per le viste materializzate in un datashare consumer quando le tabelle di base vengono condivise. L'aggiornamento incrementale è un'operazione in cui Amazon Redshift identifica le modifiche nella tabella o nelle tabelle di base avvenute dopo l'aggiornamento precedente e aggiorna solo i record corrispondenti nella vista materializzata. Per ulteriori informazioni su questo comportamento, consulta CREATE MATERIALIZED VIEW.

Limitazioni per l'aggiornamento incrementale

Amazon Redshift attualmente non supporta l'aggiornamento incrementale per le viste materializzate definite con una query utilizzando uno dei seguenti elementi SQL:

- OUTER JOIN (RIGHT, LEFT o FULL).
- Operazioni insiemistiche: UNION, INTERSECT, EXCEPT, MINUS
- UNION ALL quando si verifica in una query secondaria e una funzione di aggregazione o una clausola GROUP BY è presente nella query.
- Le funzioni di aggregazione MEDIAN, PERCENTILE_CONT, MAX, MIN, LISTAGG, STDDEV_SAMP, STDDEV_POP, APPROXIMATE COUNT, APPROXIMATE PERCENTILE e le funzioni di aggregazione bitwise.

> **Note**
>
> Sono supportate le funzioni di aggregazione COUNT, SUM, MIN, MAX e AVG.

- Funzioni di aggregazione di tipo DISTINCT, come DISTINCT COUNT, DISTINCT SUM, ecc.
- Funzioni finestra
- Query che utilizza tabelle temporanee per l'ottimizzazione delle query, ad esempio l'ottimizzazione delle espressioni secondarie comuni.
- Sottoquery
- Tabelle esterne che fanno riferimento ai seguenti formati nella query che definisce la vista materializzata.
  - Delta Lake
  - Hudi

  L'aggiornamento incrementale è supportato per le viste materializzate definite utilizzando tabelle esterne che fanno riferimento ad altri formati nella traccia di anteprima. Per ulteriori informazioni sulla configurazione dei cluster di anteprima, consulta Creazione di un cluster di anteprima nella Guida alla gestione di Amazon Redshift. Per informazioni sulla configurazione dei gruppi di lavoro di anteprima, consulta Creazione di un gruppo di lavoro di anteprima nella Guida alla gestione di Amazon Redshift.
- Funzioni mutabili, come le funzioni data-ora, funzioni RANDOM e funzioni definite dall'utente non stabili.
- Per le limitazioni relative all'aggiornamento incrementale per le integrazioni zero-ETL, consulta Considerazioni sull'utilizzo di integrazioni zero-ETL con Amazon Redshift.

Per ulteriori informazioni sulle limitazioni della vista materializzata, incluso l'effetto di operazioni in background come VACUUM sulle operazioni di aggiornamento della vista materializzata, consulta Note per l'utilizzo.

Esempi

Nell'esempio seguente viene eseguito l'aggiornamento della vista materializzata `tickets_mv`.

```
REFRESH MATERIALIZED VIEW tickets_mv;
```

RESET

Ripristina il valore predefinito di un parametro di configurazione.

Puoi ripristinare un singolo parametro specificato o tutti i parametri contemporaneamente. Per impostare un parametro su un valore specifico, utilizza il comando SET. Per visualizzare il valore corrente di un parametro, utilizza il comando MOSTRA.

Sintassi

```
RESET { parameter_name | ALL }
```

La seguente istruzione imposta il valore di una variabile di contesto di sessione su NULL.

```
RESET { variable_name | ALL }
```

Parametri

parameter_name

Nome del parametro da reimpostare. Per ulteriori informazioni sui parametri, vedi Modifica della configurazione del server.

ALL

Reimposta tutti i parametri di runtime, incluse tutte le variabili di contesto di sessione.

variabile

Il nome della variabile da reimpostare Se il valore da REIMPOSTARE è una variabile di contesto di sessione, Amazon Redshift la imposta su NULL.

Esempi

L'esempio seguente reimposta il parametro `query_group` sul valore predefinito:

```
reset query_group;
```

L'esempio seguente reimposta tutti i parametri di runtime sui valori predefiniti.

```
reset all;
```

Nell'esempio viene reimpostata la variabile di contesto.

```
RESET app_context.user_id;
```

REVOKE

Rimuove i privilegi di accesso, come i privilegi per creare, eliminare o aggiornare tabelle, da un utente o un ruolo.

Per quanto riguarda le autorizzazioni, è possibile usare solo GRANT o REVOKE USAGE su uno schema esterno per gli utenti del database e i ruoli che utilizzano la sintassi ON SCHEMA. Quando utilizzi ON EXTERNAL SCHEMA con AWS Lake Formation, puoi solo CONCEDERE e REVOCARE le autorizzazioni a un ruolo (IAM). AWS Identity and Access Management Per un elenco delle autorizzazioni richieste, consulta la sintassi.

Per le procedure archiviate, USAGE ON LANGUAGE `plpgsql` è concesso a PUBLIC per impostazione predefinita. Per impostazione predefinita, EXECUTE ON PROCEDURE è concesso solo al proprietario e agli utenti con privilegi avanzati.

Specificare nel comando REVOKE le autorizzazioni che si desidera rimuovere. Per concedere le autorizzazioni, utilizzare il comando GRANT.

Sintassi

```
REVOKE [ GRANT OPTION FOR ]
{ { SELECT | INSERT | UPDATE | DELETE | DROP | REFERENCES | ALTER | TRUNCATE } [,...] |
 ALL [ PRIVILEGES ] }
ON { [ TABLE ] table_name [, ...] | ALL TABLES IN SCHEMA schema_name [, ...] }
FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
[ RESTRICT ]


REVOKE [ GRANT OPTION FOR ]
{ { CREATE | TEMPORARY | TEMP | ALTER } [,...] | ALL [ PRIVILEGES ] }
ON DATABASE db_name [, ...]
FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
[ RESTRICT ]


REVOKE [ GRANT OPTION FOR ]
{ { CREATE | USAGE | ALTER } [,...] | ALL [ PRIVILEGES ] }
```

```
ON SCHEMA schema_name [, ...]
FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
[ RESTRICT ]


REVOKE [ GRANT OPTION FOR ]
EXECUTE
    ON FUNCTION function_name ( [ [ argname ] argtype [, ...] ] ) [, ...]
    FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
[ RESTRICT ]


REVOKE [ GRANT OPTION FOR ]
{ { EXECUTE } [,...] | ALL [ PRIVILEGES ] }
    ON PROCEDURE procedure_name ( [ [ argname ] argtype [, ...] ] ) [, ...]
    FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
[ RESTRICT ]


REVOKE [ GRANT OPTION FOR ]
USAGE
    ON LANGUAGE language_name [, ...]
    FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
[ RESTRICT ]
```

Revoca delle autorizzazioni a livello di colonna per le tabelle

Di seguito è riportata la sintassi per le autorizzazioni a livello di colonna su tabelle e viste di Amazon Redshift.

```
REVOKE { { SELECT | UPDATE } ( column_name [, ...] ) [, ...] | ALL [ PRIVILEGES ]
 ( column_name [,...] ) }
     ON { [ TABLE ] table_name [, ...] }
     FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
     [ RESTRICT ]
```

Revoca delle autorizzazioni ASSUMEROLE

Di seguito è riportata la sintassi per revocare l'autorizzazione ASSUMEROLE da utenti e gruppi con un ruolo specificato.

```
REVOKE ASSUMEROLE
    ON { 'iam_role' [, ...]  | default | ALL }
```

```
    FROM { user_name | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
    FOR { ALL | COPY | UNLOAD | EXTERNAL FUNCTION | CREATE MODEL }
```

Revoca delle autorizzazioni per Redshift Spectrum per Lake Formation

Di seguito è riportata la sintassi per l'integrazione di Redshift Spectrum con Lake Formation.

```
REVOKE [ GRANT OPTION FOR ]
{ SELECT | ALL [ PRIVILEGES ] } ( column_list )
    ON EXTERNAL TABLE schema_name.table_name
    FROM { IAM_ROLE iam_role } [, ...]

REVOKE [ GRANT OPTION FOR ]
{ { SELECT | ALTER | DROP | DELETE | INSERT }  [, ...] | ALL [ PRIVILEGES ] }
    ON EXTERNAL TABLE schema_name.table_name [, ...]
    FROM { { IAM_ROLE iam_role } [, ...] | PUBLIC }

REVOKE [ GRANT OPTION FOR ]
{ { CREATE | ALTER | DROP }  [, ...] | ALL [ PRIVILEGES ] }
    ON EXTERNAL SCHEMA schema_name [, ...]
    FROM { IAM_ROLE iam_role } [, ...]
```

Revoca delle autorizzazioni dell'unità di condivisione dati

Autorizzazioni dell'unità di condivisione dati sul lato producer

Di seguito è riportata la sintassi per l'utilizzo di REVOKE per rimuovere le autorizzazioni ALTER o SHARE da un utente o un ruolo. L'utente con le autorizzazioni revocate non può più modificare l'unità di condivisione dati o assegnarne l'utilizzo a un consumer.

```
REVOKE { ALTER | SHARE } ON DATASHARE datashare_name
 FROM { username [ WITH GRANT OPTION ] | ROLE role_name | GROUP group_name | PUBLIC }
 [, ...]
```

Di seguito è riportata la sintassi per l'utilizzo di REVOKE per rimuovere l'accesso di un utente a un'unità di condivisione dati.

```
REVOKE USAGE
 ON DATASHARE datashare_name
 FROM NAMESPACE 'namespaceGUID' [, ...] | ACCOUNT 'accountnumber' [ VIA DATA CATALOG ]
 [, ...]
```

Di seguito è riportato un esempio per revocare l'autorizzazione di utilizzo di un'unità di condivisione dati a un account Lake Formation.

```
REVOKE USAGE ON DATASHARE salesshare FROM ACCOUNT '123456789012' VIA DATA CATALOG;
```

Autorizzazioni dell'unità di condivisione dati sul lato consumer

Di seguito è riportata la sintassi di REVOKE per le autorizzazioni di utilizzo per la condivisione dei dati su un database o uno schema specifico creato da una unità di condivisione dati. La revoca dell'autorizzazione di utilizzo a un database creato con la clausola WITH PERMISSIONS non revoca le autorizzazioni aggiuntive assegnate a un utente o a un ruolo, incluse quelle a livello di oggetto assegnate agli oggetti sottostanti. Se assegni nuovamente l'autorizzazione di utilizzo all'utente o al ruolo, verranno mantenute tutte le autorizzazioni aggiuntive di cui l'utente o il ruolo disponeva prima della revoca dell'utilizzo.

```
REVOKE USAGE ON { DATABASE shared_database_name [, ...] | SCHEMA shared_schema}
 FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
```

Revoca delle autorizzazioni con ambito

Le autorizzazioni con ambito consentono di concedere autorizzazioni a un utente o a un ruolo su tutti gli oggetti di un tipo all'interno di un database o di uno schema. Gli utenti e i ruoli con autorizzazioni mirate dispongono delle autorizzazioni specificate su tutti gli oggetti attuali e futuri all'interno del database o dello schema.

Di seguito è riportata la sintassi per revocare a utenti e ruoli autorizzazioni specifiche. Per ulteriori informazioni sulle autorizzazioni con ambito, vedere. Autorizzazioni con ambito

```
REVOKE [ GRANT OPTION ]
{ CREATE | USAGE | ALTER } [,...] | ALL [ PRIVILEGES ] }
FOR SCHEMAS IN
DATABASE db_name
FROM { username | ROLE role_name } [, ...]

REVOKE [ GRANT OPTION ]
{ { SELECT | INSERT | UPDATE | DELETE | DROP | ALTER | TRUNCATE | REFERENCES }
 [, ...] } | ALL [PRIVILEGES] } }
FOR TABLES IN
{SCHEMA schema_name [DATABASE db_name ] | DATABASE db_name }
FROM { username] | ROLE role_name} [, ...]
```

```
REVOKE [ GRANT OPTION ] { EXECUTE | ALL [ PRIVILEGES ] }
FOR FUNCTIONS IN
{SCHEMA schema_name [DATABASE db_name ] | DATABASE db_name }
FROM { username | ROLE role_name | } [, ...]

REVOKE [ GRANT OPTION ] { EXECUTE | ALL [ PRIVILEGES ] }
FOR PROCEDURES IN
{SCHEMA schema_name [DATABASE db_name ] | DATABASE db_name }
FROM { username | ROLE role_name | } [, ...]

REVOKE [ GRANT OPTION ] USAGE
FOR LANGUAGES IN
{DATABASE db_name}
FROM { username | ROLE role_name } [, ...]
```

Tieni presente che le autorizzazioni con ambito non fanno distinzione tra le autorizzazioni per le funzioni e per le procedure. Ad esempio, l'istruzione seguente revoca le EXECUTE autorizzazioni sia per le funzioni che per le procedure dallo schema. bob `Sales_schema`

```
REVOKE EXECUTE FOR FUNCTIONS IN SCHEMA Sales_schema FROM bob;
```

Revoca delle autorizzazioni di machine learning

Di seguito è riportata la sintassi per le autorizzazioni per il modello di machine learning su Amazon Redshift.

```
REVOKE [ GRANT OPTION FOR ]
    CREATE MODEL FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
    [ RESTRICT ]

REVOKE [ GRANT OPTION FOR ]
    { EXECUTE | ALL [ PRIVILEGES ] }
    ON MODEL model_name [, ...]

    FROM { username | ROLE role_name | GROUP group_name | PUBLIC } [, ...]
    [ RESTRICT ]
```

Revoca delle autorizzazioni per i ruoli

Di seguito è riportata la sintassi per revocare le autorizzazioni per i ruoli su Amazon Redshift.

```
REVOKE [ ADMIN OPTION FOR ] { ROLE role_name } [, ...] FROM { user_name } [, ...]
```

```
REVOKE { ROLE role_name } [, ...] FROM { ROLE role_name } [, ...]
```

Di seguito è riportata la sintassi per revocare le autorizzazioni di sistema per i ruoli su Amazon Redshift.

```
REVOKE
  {
    { CREATE USER | DROP USER | ALTER USER |
    CREATE SCHEMA | DROP SCHEMA |
    ALTER DEFAULT PRIVILEGES |
    ACCESS CATALOG |
    CREATE TABLE | DROP TABLE | ALTER TABLE |
    CREATE OR REPLACE FUNCTION | CREATE OR REPLACE EXTERNAL FUNCTION |
    DROP FUNCTION |
    CREATE OR REPLACE PROCEDURE | DROP PROCEDURE |
    CREATE OR REPLACE VIEW | DROP VIEW |
    CREATE MODEL | DROP MODEL |
    CREATE DATASHARE | ALTER DATASHARE | DROP DATASHARE |
    CREATE LIBRARY | DROP LIBRARY |
    CREATE ROLE | DROP ROLE
    TRUNCATE TABLE
    VACUUM | ANALYZE | CANCEL }[, ...]
  }
  | { ALL [ PRIVILEGES ] }
FROM { ROLE role_name } [, ...]
```

Revoca delle autorizzazioni esplicative per i filtri delle policy di sicurezza a livello di riga

Di seguito è riportata la sintassi per la revoca delle autorizzazioni che spiega i filtri delle policy di sicurezza a livello di riga di una query nel piano EXPLAIN. Puoi revocare il privilegio utilizzando l'istruzione REVOKE.

```
REVOKE EXPLAIN RLS FROM ROLE rolename
```

Di seguito è riportata la sintassi per concedere le autorizzazioni per aggirare le policy di sicurezza a livello di riga per una query.

```
REVOKE IGNORE RLS FROM ROLE rolename
```

Di seguito è riportata la sintassi per revocare le autorizzazioni dalla policy di sicurezza a livello di riga.

```
REVOKE SELECT ON [ TABLE ] table_name [, ...]
            FROM RLS POLICY policy_name [, ...]
```

Parametri

GRANT OPTION FOR

Revoca solo l'opzione per concedere un'autorizzazione specificata ad altri utenti e non revoca l'autorizzazione stessa. Non puoi revocare GRANT OPTION a un gruppo o a PUBLIC.

SELECT

Revoca l'autorizzazione a selezionare i dati da una tabella o una vista utilizzando un'istruzione SELECT.

INSERT

Revoca l'autorizzazione a caricare i dati in una tabella utilizzando un'istruzione INSERT o un'istruzione COPY.

UPDATE

Revoca l'autorizzazione ad aggiornare una colonna della tabella utilizzando un'istruzione UPDATE.

DELETE

Revoca l'autorizzazione a eliminare una riga di dati da una tabella.

REFERENCES

Revoca l'autorizzazione a creare un vincolo di chiave esterna. È necessario revocare questa autorizzazione sia sulla tabella a cui si fa riferimento, sia sulla tabella che fa da riferimento.

TRUNCATE

Revoca l'autorizzazione a troncare una tabella. Senza questa autorizzazione, solo il proprietario di una tabella o un utente con privilegi avanzati può troncare una tabella. Per ulteriori informazioni sul comando TRUNCATE, consulta the section called “TRUNCATE”.

ALL [PRIVILEGES]

Revoca contemporaneamente tutte le autorizzazioni disponibili per l'utente o il gruppo specificato. La parola chiave PRIVILEGES è facoltativa.

ALTER

A seconda dell'oggetto del database, revoca le seguenti autorizzazioni all'utente o al gruppo di utenti:

- Per le tabelle, ALTER revoca l'autorizzazione a modificare una tabella o una vista. Per ulteriori informazioni, consulta ALTER TABLE.
- Per i database, ALTER revoca l'autorizzazione a modificare un database. Per ulteriori informazioni, consulta ALTER DATABASE.
- Per gli schemi, ALTER revoca l'autorizzazione a modificare uno schema. Per ulteriori informazioni, consulta ALTER SCHEMA.
- Per le tabelle esterne, ALTER revoca l'autorizzazione a modificare una tabella in un AWS Glue Data Catalog file abilitato per Lake Formation. Questa autorizzazione si applica solo quando si utilizza Lake Formation.

DROP

Revoca l'autorizzazione a eliminare una tabella. Questa autorizzazione si applica in Amazon Redshift e in un AWS Glue Data Catalog ambiente abilitato per Lake Formation.

ASSUMEROLE

Revoca l'autorizzazione a eseguire i comandi COPY, UNLOAD, EXTERNAL FUNCTION o CREATE MODEL per utenti, ruoli o gruppi con un ruolo specificato.

ON [TABLE] table_name

Revoca le autorizzazioni specificate su una tabella o una vista. La parola chiave TABLE è facoltativa.

ON ALL TABLES IN SCHEMA schema_name

Revoca le autorizzazioni specificate su tutte le tabelle nello schema a cui si fa riferimento.

(column_name [,...]) ON TABLE table_name

Revoca le autorizzazioni specificate da utenti, gruppi o PUBLIC nelle colonne specificate della tabella o della vista Amazon Redshift.

(column_list) ON EXTERNAL TABLE schema_name.table_name

Revoca le autorizzazioni specificate da un ruolo IAM nelle colonne indicate della tabella Lake Formation nello schema a cui si fa riferimento.

ON EXTERNAL TABLE schema_name.table_name

Revoca le autorizzazioni specificate da un ruolo IAM nelle tabelle Lake Formation specificate nello schema a cui si fa riferimento.

ON EXTERNAL SCHEMA schema_name

Revoca le autorizzazioni specificate da un ruolo IAM nello schema a cui si fa riferimento.

FROM IAM_ROLE iam_role

Indica il ruolo IAM che perde le autorizzazioni.

ROLE role_name

Revoca le autorizzazioni dal ruolo specificato.

GROUP group_name

Revoca le autorizzazioni al gruppo di utenti specificato.

PUBLIC

Revoca le autorizzazioni a tutti gli utenti. PUBLIC rappresenta un gruppo che include sempre tutti gli utenti. Le autorizzazioni di un singolo utente consistono nella somma delle autorizzazioni concesse a PUBLIC, delle autorizzazioni concesse a tutti i gruppi a cui l'utente appartiene e di tutte le autorizzazioni concesse all'utente singolarmente.

La revoca di PUBLIC da una tabella esterna di Lake Formation comporta la revoca dell'autorizzazione al gruppo everyone di Lake Formation.

CREATE

A seconda dell'oggetto del database, revoca le seguenti autorizzazioni all'utente o al gruppo:

- Per i database, l'utilizzo della clausola CREATE per REVOKE impedisce agli utenti di creare schemi all'interno del database.
- Per gli schemi, l'utilizzo della clausola CREATE per REVOKE impedisce agli utenti di creare oggetti all'interno di uno schema. Per rinominare un oggetto, l'utente deve avere l'autorizzazione CREATE ed essere il proprietario dell'oggetto da rinominare.

> **Note**
>
> Di default, tutti gli utenti dispongono delle autorizzazioni CREATE e USAGE per lo schema PUBLIC di un database.

TEMPORARY | TEMP

Revoca l'autorizzazione a creare tabelle temporanee nel database specificato.

> **Note**
>
> Per impostazione predefinita, agli utenti è concessa l'autorizzazione di creare tabelle temporanee tramite la loro appartenenza automatica al gruppo PUBLIC. Per rimuovere l'autorizzazione per qualsiasi utente di creare tabelle temporanee, revoca l'autorizzazione TEMP dal gruppo PUBLIC e quindi concedi esplicitamente l'autorizzazione per creare tabelle temporanee per utenti specifici o gruppi di utenti.

ON DATABASE db_name

Revoca le autorizzazioni sul database specificato.

USAGE

Revoca le autorizzazioni USAGE sugli oggetti all'interno di uno schema specifico, rendendo questi oggetti inaccessibili agli utenti. Le operazioni specifiche su questi oggetti devono essere revocate separatamente (come l'autorizzazione EXECUTE sulle funzioni).

> **Note**
>
> Di default, tutti gli utenti dispongono delle autorizzazioni CREATE e USAGE per lo schema PUBLIC di un database.

ON SCHEMA schema_name

Revoca le autorizzazioni sullo schema specificato. Puoi utilizzare le autorizzazioni dello schema per controllare la creazione delle tabelle; l'autorizzazione CREATE per un database controlla solo la creazione degli schemi.

RESTRICT

Revoca solo le autorizzazioni che l'utente ha concesso direttamente. Questo comportamento è quello predefinito.

EXECUTE ON PROCEDURE procedure_name

Revoca l'autorizzazione EXECUTE su una procedura archiviata specifica. Poiché i nomi delle procedure archiviate possono essere in overload, devi includere l'elenco degli argomenti per la procedura. Per ulteriori informazioni, consulta Denominazione delle stored procedure.

EXECUTE ON ALL PROCEDURES IN SCHEMA procedure_name

Revoca le autorizzazioni specificate su tutte le procedure nello schema a cui si fa riferimento.

USAGE ON LANGUAGE language_name

Revoca l'autorizzazione USAGE per una lingua. Per le funzioni definite dall'utente (UDF) Python, usa `plpythonu`. Per le funzioni definite dall'utente SQL usa `sql`. Per le procedure archiviate, usa `plpgsql`.

Per creare una UDF devi disporre dell'autorizzazione per l'utilizzo nel linguaggio per SQL o `plpythonu` (Python). Per impostazione predefinita, USAGE ON LANGUAGE SQL è concesso a PUBLIC. Tuttavia, devi concedere esplicitamente USAGE ON LANGUAGE PLPYTHONU a utenti o gruppi specifici.

Per revocare l'utilizzo per SQL, revoca innanzitutto l'utilizzo da PUBLIC. Quindi concedi l'utilizzo su SQL solo a utenti o gruppi specifici autorizzati a creare UDF SQL. L'esempio seguente revoca l'utilizzo su SQL da PUBLIC, quindi concede l'utilizzo al gruppo di utenti `udf_devs`.

```
revoke usage on language sql from PUBLIC;
grant usage on language sql to group udf_devs;
```

Per ulteriori informazioni, consulta Sicurezza e privilegi dell'UDF.

Per revocare l'utilizzo per le procedure archiviate, revoca innanzitutto l'utilizzo da PUBLIC. Quindi concedi l'utilizzo su `plpgsql` solo a utenti o gruppi specifici autorizzati a creare procedure archiviate. Per ulteriori informazioni, consulta Sicurezza e privilegi per le procedure archiviate .

FOR { ALL | COPY | UNLOAD | EXTERNAL FUNCTION | CREATE MODEL } [, ...]

Specifica il comando SQL per il quale viene revocata l'autorizzazione. Puoi specificare ALL per revocare l'autorizzazione sulle istruzioni COPY, UNLOAD, EXTERNAL FUNCTION e CREATE MODEL. Questa clausola si applica solo alla revoca dell'autorizzazione ASSUMEROLE.

ALTER

Revoca l'autorizzazione ALTER per utenti o gruppi di utenti che consente a coloro che non possiedono una unità di condivisione dati di modificarla. Questa autorizzazione è richiesta per

aggiungere o rimuovere oggetti da una unità di condivisione dati o per impostare la proprietà PUBLICACCESSIBLE. Per ulteriori informazioni, consulta ALTER DATASHARE.

SHARE

Revoca le autorizzazioni a utenti e gruppi di utenti per aggiungere consumer a una unità di condivisione dati. La revoca di questa autorizzazione è necessaria per impedire al particolare consumer di accedere all'unità di condivisione dati dai suoi cluster.

ON DATASHARE nome_unità_condivisione_dati

Concede le autorizzazioni specificate sull'unità di condivisione dati a cui si fa riferimento.

FROM username

Indica il ruolo IAM che perde le autorizzazioni.

FROM GROUP nome_gruppo

Indica il gruppo di utenti che perde le autorizzazioni.

WITH GRANT OPTION

Indica che l'utente che perde le autorizzazioni può a sua volta revocare le stesse autorizzazioni ad altri. Non è possibile revocare WITH GRANT OPTION per un gruppo o per PUBLIC.

USAGE

Quando USAGE viene revocato per un account consumer o uno spazio dei nomi all'interno dello stesso account, l'account consumer specifico o lo spazio dei nomi all'interno dell'account non potrà accedere all'unità di condivisione dati e gli oggetti dell'unità di condivisione dati saranno in sola lettura.

La revoca dell'autorizzazione USAGE revoca l'accesso a una unità di condivisione dati da parte dei consumer.

FROM NAMESPACE 'clusternamespace GUID'

Indica lo spazio dei nomi nello stesso account in cui i consumer perdono le autorizzazioni per l'unità di condivisione dati. Gli spazi dei nomi utilizzano un identificatore univoco globale (GUID) alfanumerico a 128 bit.

FROM ACCOUNT 'accountnumber' [VIA DATA CATALOG]

Indica il numero di account di un altro account in cui i consumer perdono le autorizzazioni per l'unità di condivisione dati. Specificare 'VIA DATA CATALOG' indica che si sta revocando

l'autorizzazione per l'utilizzo del'unità di condivisione dati da un account Lake Formation. L'omissione del numero di account indica che lo si sta revocando dall'account proprietario del cluster.

ON DATABASE nome_database_condiviso> [, ...]

Revoca le autorizzazioni di utilizzo indicate per il database specificato creato nell'unità di condivisione dati specificata.

ON SCHEMA schema_condiviso

Revoca le autorizzazioni indicate per lo schema specificato creato nell'unità di condivisione dati specificata.

FOR { SCHEMAS | TABLES | FUNCTIONS | PROCEDURES | LANGUAGES } IN

Specifica gli oggetti del database a cui revocare l'autorizzazione. I parametri successivi a IN definiscono l'ambito dell'autorizzazione revocata.

CREATE MODEL

Revoca l'autorizzazione CREATE MODEL per creare modelli di machine learning nel database specificato.

ON MODEL nome_modello

Revoca l'autorizzazione EXECUTE per un modello specifico.

ACCESS CATALOG

Revoca l'autorizzazione a visualizzare i metadati pertinenti degli oggetti a cui il ruolo ha accesso.

[ADMIN OPTION FOR] { role } [, ...]

Il ruolo che si revoca da un utente specificato che dispone dell'opzione WITH ADMIN OPTION.

FROM { role } [, ...]

Il ruolo da cui si revoca il ruolo specificato.

Note per l'utilizzo

Per ulteriori informazioni sulle note di utilizzo di REVOKE, consulta the section called “Note per l'utilizzo”.

Esempi

Per gli esempi di come utilizzare REVOKE, consulta the section called “Esempi”.

Note per l'utilizzo

Per revocare i privilegi a un oggetto, è necessario soddisfare uno dei seguenti criteri:

- Essere il proprietario dell'oggetto.
- Essere un utente con privilegi avanzati.
- Avere un privilegio di concessione per l'oggetto e il privilegio.

  Ad esempio, il seguente comando consente all'utente HR di eseguire i comandi SELECT sulla tabella dei dipendenti e di concedere e revocare lo stesso privilegio per altri utenti.

  ```
  grant select on table employees to HR with grant option;
  ```

  HR non può revocare privilegi per operazioni diverse da SELECT o su qualsiasi altra tabella rispetto a quella dei dipendenti.

Gli utenti con privilegi avanzati possono accedere a tutti gli oggetti indipendentemente dai comandi GRANT e REVOKE che impostano i privilegi dell'oggetto.

PUBLIC rappresenta un gruppo che include sempre tutti gli utenti. Per impostazione predefinita tutti i membri di PUBLIC hanno i privilegi CREATE e USAGE nello schema PUBLIC. Per limitare le autorizzazioni di qualsiasi utente sullo schema PUBLIC, è necessario prima revocare tutte le autorizzazioni da PUBLIC sullo schema PUBLIC, quindi concedere i privilegi a specifici utenti o gruppi. L'esempio seguente controlla i privilegi di creazione della tabella nello schema PUBLIC.

```
revoke create on schema public from public;
```

Per revocare i privilegi da una tabella Lake Formation, il ruolo IAM associato con lo schema esterno della tabella deve avere l'autorizzazione per revocare i privilegi alla tabella esterna. L'esempio seguente crea uno schema esterno con un ruolo IAM associato `myGrantor`. Il ruolo IAM `myGrantor` ha l'autorizzazione di revocare le autorizzazioni da altri. Il comando REVOKE utilizza l'autorizzazione del ruolo IAM `myGrantor` associato allo schema esterno per revocare l'autorizzazione al ruolo IAM `myGrantee`.

```
create external schema mySchema
from data catalog
database 'spectrum_db'
iam_role 'arn:aws:iam::123456789012:role/myGrantor'
create external database if not exists;
```

```
revoke select
on external table mySchema.mytable
from iam_role 'arn:aws:iam::123456789012:role/myGrantee';
```

> **Note**
>
> Se il ruolo IAM dispone anche dell'`ALL`autorizzazione in un AWS Glue Data Catalog account abilitato per Lake Formation, l'`ALL`autorizzazione non viene revocata. Viene revocata solo autorizzazione `SELECT`. È possibile visualizzare le autorizzazioni Lake Formation nella console Lake Formation.

Note di utilizzo per la revoca dell'autorizzazione ASSUMEROLE

Le seguenti note di utilizzo si applicano alla revoca del privilegio ASSUMEROLE in Amazon Redshift.

Solo un utente con privilegi avanzati del database può revocare il privilegio ASSUMEROLE per utenti e gruppi. Un utente con privilegi avanzati mantiene sempre il privilegio ASSEMEROLE.

Per abilitare l'uso del privilegio ASSUMEROLE per utenti e gruppi, un utente con privilegi avanzati esegue l'istruzione riportata una volta nel cluster. Prima di concedere il privilegio ASSUMEROLE a utenti e gruppi, un utente con privilegi avanzati deve eseguire l'istruzione riportata una volta nel cluster.

```
revoke assumerole on all from public for all;
```

Note di utilizzo per la revoca delle autorizzazioni di machine learning

Non è possibile concedere o revocare direttamente le autorizzazioni relative a una funzione ML. Una funzione ML appartiene a un modello ML e le autorizzazioni sono controllate tramite il modello. È invece possibile revocare le autorizzazioni relative al modello ML. L'esempio seguente dimostra come revocare le autorizzazioni di esecuzione a tutti gli utenti associati al modello `customer_churn`.

```
REVOKE EXECUTE ON MODEL customer_churn FROM PUBLIC;
```

Puoi anche revocare tutte le autorizzazioni a un utente per il modello ML `customer_churn`.

```
REVOKE ALL on MODEL customer_churn FROM ml_user;
```

La concessione o la revoca dell'autorizzazione `EXECUTE` relativa a una funzione ML avrà esito negativo se nello schema è presente una funzione ML, anche se tale funzione ML dispone già dell'autorizzazione `EXECUTE` tramite `GRANT EXECUTE ON MODEL`. Si consiglia di utilizzare uno schema separato quando si utilizza il comando `CREATE MODEL` per mantenere le funzioni ML in uno schema separato. L'esempio seguente mostra come fare.

```
CREATE MODEL ml_schema.customer_churn
FROM customer_data
TARGET churn
FUNCTION ml_schema.customer_churn_prediction
IAM_ROLE default
SETTINGS (
 S3_BUCKET 'your-s3-bucket'
);
```

Esempi

L'esempio seguente revoca i privilegi INSERT sulla tabella SALES dal gruppo utenti GUESTS. Questo comando impedisce ai membri di GUESTS di caricare i dati nella tabella SALES utilizzando il comando INSERT.

```
revoke insert on table sales from group guests;
```

L'esempio seguente revoca all'utente il privilegio SELECT su tutte le tabelle dello schema QA_TICKIT: `fred`.

```
revoke select on all tables in schema qa_tickit from fred;
```

L'esempio seguente revoca all'utente il privilegio di selezionare da una vista: `bobr`.

```
revoke select on table eventview from bobr;
```

L'esempio seguente revoca il privilegio di creare tabelle temporanee nel database TICKIT a tutti gli utenti:

```
revoke temporary on database tickit from public;
```

L'esempio seguente revoca il privilegio SELECT sulle colonne `cust_name` e `cust_phone` della tabella `cust_profile` all'utente `user1`.

```
revoke select(cust_name, cust_phone) on cust_profile from user1;
```

L'esempio seguente revoca il privilegio SELECT sulle colonne `cust_name` e `cust_phone` e il privilegio UPDATE sulla colonna `cust_contact_preference` della tabella `cust_profile` dal gruppo `sales_group`.

```
revoke select(cust_name, cust_phone), update(cust_contact_preference) on cust_profile
 from group sales_group;
```

Nell'esempio seguente viene illustrato l'utilizzo della parola chiave ALL per revocare i privilegi SELECT e UPDATE su tre colonne della tabella `cust_profile` dal gruppo `sales_admin`.

```
revoke ALL(cust_name, cust_phone,cust_contact_preference) on cust_profile from group
 sales_admin;
```

L'esempio seguente revoca il privilegio SELECT nella colonna `cust_name` della vista `cust_profile_vw` dall'utente `user2`.

```
revoke select(cust_name) on cust_profile_vw from user2;
```

Esempi di revoca dell'autorizzazione USAGE ai database creati da unità di condivisione dati

L'esempio seguente revoca l'accesso all'unità di condivisione dati `salesshare` allo spazio dei nomi `13b8833d-17c6-4f16-8fe4-1a018f5ed00d`.

```
REVOKE USAGE ON DATASHARE salesshare FROM NAMESPACE
 '13b8833d-17c6-4f16-8fe4-1a018f5ed00d';
```

L'esempio seguente revoca l'autorizzazione USAGE per `sales_db` a `Bob`.

```
REVOKE USAGE ON DATABASE sales_db FROM Bob;
```

Nell'esempio seguente REVOKE USAGE revoca l'autorizzazione per `sales_schema` a `Analyst_role`.

```
REVOKE USAGE ON SCHEMA sales_schema FROM ROLE Analyst_role;
```

Esempi di revoca di autorizzazioni con ambito

L'esempio seguente revoca al ruolo `Sales` l'utilizzo di tutti gli schemi attuali e futuri nel database `Sales_db`.

```
REVOKE USAGE FOR SCHEMAS IN DATABASE Sales_db FROM ROLE Sales;
```

L'esempio seguente revoca la possibilità di concedere all'utente `alice` l'autorizzazione SELECT per tutte le tabelle correnti e future del database `Sales_db`. `alice` mantiene l'accesso a tutte le tabelle in `Sales_db`.

```
REVOKE GRANT OPTION SELECT FOR TABLES IN DATABASE Sales_db FROM alice;
```

L'esempio seguente revoca all'utente bob l'autorizzazione EXECUTE per le funzioni dello schema `Sales_schema`.

```
REVOKE EXECUTE FOR FUNCTIONS IN SCHEMA Sales_schema FROM bob;
```

L'esempio seguente revoca al ruolo `Sales` tutte le autorizzazioni per tutte le tabelle dello schema `ShareSchema` del database `ShareDb`. Quando si specifica lo schema, è anche possibile indicare il database dello schema utilizzando il formato in due parti `database.schema`.

```
REVOKE ALL FOR TABLES IN SCHEMA ShareDb.ShareSchema FROM ROLE Sales;
```

L'esempio seguente è uguale al precedente. Puoi specificare il database dello schema utilizzando la parola chiave DATABASE anziché utilizzare un formato in due parti.

```
REVOKE ALL FOR TABLES IN SCHEMA ShareSchema DATABASE ShareDb FROM ROLE Sales;
```

Esempi di revoca del privilegio ASSUMEROLE

Di seguito sono riportati esempi di revoca del privilegio ASSUMEROLE.

Un utente con privilegi avanzati deve abilitare l'uso del privilegio ASSUMEROLE per utenti e gruppi eseguendo l'istruzione riportata una volta nel cluster.

```
revoke assumerole on all from public for all;
```

L'istruzione seguente revoca il privilegio ASSUMEROLE dall'utente reg_user1 su tutti i ruoli per tutte le operazioni.

```
revoke assumerole on all from reg_user1 for all;
```

Esempi di revoca del privilegio ROLE

L'esempio seguente revoca sample_role1 da sample_role2.

```
CREATE ROLE sample_role2;
GRANT ROLE sample_role1 TO ROLE sample_role2;
REVOKE ROLE sample_role1 FROM ROLE sample_role2;
```

L'esempio seguente revoca i privilegi di sistema da user1.

```
GRANT ROLE sys:DBA TO user1;
REVOKE ROLE sys:DBA FROM user1;
```

L'esempio seguente revoca sample_role1 e sample_role2 da user1.

```
CREATE ROLE sample_role1;
CREATE ROLE sample_role2;
GRANT ROLE sample_role1, ROLE sample_role2 TO user1;
REVOKE ROLE sample_role1, ROLE sample_role2 FROM user1;
```

L'esempio seguente revoca sample_role2 con l'opzione ADMIN OPTION da user1.

```
GRANT ROLE sample_role2 TO user1 WITH ADMIN OPTION;
REVOKE ADMIN OPTION FOR ROLE sample_role2 FROM user1;
REVOKE ROLE sample_role2 FROM user1;
```

L'esempio seguente revoca sample_role1 e sample_role2 da sample_role5.

```
CREATE ROLE sample_role5;
```

```
GRANT ROLE sample_role1, ROLE sample_role2 TO ROLE sample_role5;
REVOKE ROLE sample_role1, ROLE sample_role2 FROM ROLE sample_role5;
```

Nell'esempio seguente vengono revocati i privilegi di sistema CREATE SCHEMA e DROP SCHEMA a sample_role1.

```
GRANT CREATE SCHEMA, DROP SCHEMA TO ROLE sample_role1;
REVOKE CREATE SCHEMA, DROP SCHEMA FROM ROLE sample_role1;
```

ROLLBACK

Interrompe la transazione corrente ed elimina tutti gli aggiornamenti apportati da quella transazione.

Questo comando esegue la stessa funzione del comando ABORT.

Sintassi

```
ROLLBACK [ WORK | TRANSACTION ]
```

Parametri

WORK

Parola chiave facoltativa. Questa parola chiave non è supportata all'interno di una procedura archiviata.

TRANSACTION

Parola chiave facoltativa. WORK e TRANSACTION sono sinonimi. Nessuno dei due è supportato all'interno d una procedura archiviata.

Per informazioni sull'utilizzo di ROLLBACK all'interno di una procedura guidata, consultare Gestione delle transazioni.

Esempio

L'esempio seguente crea una tabella quindi avvia una transazione in cui i dati vengono inseriti nella tabella. Il comando ROLLBACK esegue quindi il rollback dell'inserimento dei dati per lasciare vuota la tabella.

Il seguente comando crea una tabella di esempio chiamata MOVIE_GROSS:

```
create table movie_gross( name varchar(30), gross bigint );
```

Il prossimo set di comandi avvia una transazione che inserisce due righe di dati nella tabella:

```
begin;

insert into movie_gross values ( 'Raiders of the Lost Ark', 23400000);

insert into movie_gross values ( 'Star Wars', 10000000 );
```

Successivamente, il seguente comando seleziona i dati dalla tabella per mostrare che è stato inserito:

```
select * from movie_gross;
```

L'output del comando mostra che entrambe le righe sono inserite:

```
name           |  gross
-------------------------+----------
Raiders of the Lost Ark | 23400000
Star Wars               | 10000000
(2 rows)
```

Questo comando esegue ora il rollback delle modifiche dei dati dove è iniziata la transazione:

```
rollback;
```

Selezionando i dati dalla tabella ora mostra una tabella vuota:

```
select * from movie_gross;

name | gross
------+-------
(0 rows)
```

SELECT

Restituisce le righe da tabelle, viste e funzioni definite dall'utente.

> **Note**
>
> Le dimensioni massime per una istruzione SQL è di 16 MB.

Sintassi

```
[ WITH with_subquery [, ...] ]
SELECT
[ TOP number | [ ALL | DISTINCT ]
* | expression [ AS output_name ] [, ...] ]
[ FROM table_reference [, ...] ]
[ WHERE condition ]
[ [ START WITH expression ] CONNECT BY expression ]
[ GROUP BY expression [, ...] ]
[ HAVING condition ]
[ QUALIFY condition ]
[ { UNION | ALL | INTERSECT | EXCEPT | MINUS } query ]
[ ORDER BY expression [ ASC | DESC ] ]
[ LIMIT { number | ALL } ]
[ OFFSET start ]
```

Argomenti

- Clausola WITH
- Elenco SELECT
- Clausola FROM
- Clausola WHERE
- Clausola GROUP BY
- Clausola HAVING
- Clausola QUALIFY
- UNION, INTERSECT ed EXCEPT
- Clausola ORDER BY
- Clausola CONNECT BY
- Esempi di sottoquery
- Sottoquery correlate

Clausola WITH

Una clausola WITH è una clausola facoltativa che precede l'elenco SELECT in una query. La clausola WITH definisce uno o più common_table_expression. Ogni espressione comune di tabella (CTE) definisce una tabella temporanea, che è simile a una definizione di vista. È possibile fare riferimento a queste tabelle temporanee nella clausola FROM. Vengono utilizzati solo durante l'esecuzione della query a cui appartengono. Ogni CTE nella clausola WITH specifica un nome di tabella, un elenco facoltativo di nomi di colonna e un'espressione di query che restituisce una tabella (un'istruzione SELECT). Quando si fa riferimento al nome della tabella temporanea nella clausola FROM della stessa espressione di query che la definisce, la CTE è ricorsiva.

Le sottoquery della clausola WITH sono un modo efficace per definire le tabelle che possono essere utilizzate durante l'esecuzione di una singola query. In ogni caso, è possibile ottenere gli stessi risultati utilizzando le sottoquery nel corpo principale dell'istruzione SELECT, ma le sottoquery della clausola WITH potrebbero essere più semplici da scrivere e leggere. Ove possibile, le sottoquery della clausola WITH che sono referenziate più volte sono ottimizzate come sottoespressioni comuni; vale a dire, potrebbe essere possibile valutare una sottoquery WITH una volta e riutilizzarne i risultati. Tieni presente che le sottoespressioni comuni non sono limitate a quelle definite nella clausola WITH.

Sintassi

```
[ WITH [RECURSIVE] common_table_expression [, common_table_expression , ...] ]
```

dove common_table_expression può essere non ricorsivo o ricorsivo. Di seguito è riportata la forma non ricorsiva:

```
CTE_table_name [ ( column_name [, ...] ) ] AS ( query )
```

Di seguito è riportata la forma ricorsiva di common_table_expression:

```
CTE_table_name (column_name [, ...] ) AS ( recursive_query )
```

Parametri

RECURSIVE

Parola chiave che identifichi la query come CTE ricorsiva. Questa parola chiave è obbligatoria se common_table_expression definito nella clausola WITH è ricorsivo. È possibile specificare la parola chiave RICURSIVE una sola volta, immediatamente dopo la parola chiave WITH,

anche quando la clausola WITH contiene più CTE ricorsive. In generale, una CTE ricorsiva è una sottoquery UNION ALL con due parti.

common_table_expression

Definisce una tabella temporanea a cui è possibile fare riferimento nella Clausola FROM e viene utilizzato solo durante l'esecuzione della query a cui appartiene.

CTE_table_name

Nome univoco per una tabella temporanea che definisce i risultati di una sottoquery della clausola WITH. Non puoi utilizzare nomi duplicati in una singola clausola WITH. A ogni sottoquery deve essere assegnato un nome di tabella a cui è possibile fare riferimento nella Clausola FROM.

column_name

Un elenco facoltativo di nomi di colonna di output per la sottoquery della clausola WITH, separati da virgole. Il numero dei nomi di colonna specificati deve essere uguale o inferiore al numero delle colonne definite dalla sottoquery. Per una CTE non ricorsiva, column_name è facoltativo. Per una CTE ricorsiva, column_name è obbligatorio.

query

Qualsiasi query SELECT supportata da Amazon Redshift. Per informazioni, consultare SELECT.

recursive_query

Una query UNION ALL costituita da due sottoquery SELECT:

- La prima sottoquery SELECT non ha un riferimento ricorsivo allo stesso CTE_table_name. Restituisce un set di risultati che è il seed iniziale della ricorsione. Questa parte è chiamata membro iniziale o membro del seed.
- La seconda sottoquery SELECT fa riferimento allo stesso CTE_table_name nella sua clausola FROM. Questo è chiamato membro ricorsivo. La recursive_query contiene una condizione WHERE per terminare la ecursive_query.

Note per l'utilizzo

Puoi utilizzare una clausola WITH nelle seguenti istruzioni SQL:

- SELECT
- SELECT INTO
- CREATE TABLE AS

- CREATE VIEW
- DECLARE
- EXPLAIN
- INSERT INTO...SELECT
- PREPARE
- UPDATE (all'interno di una sottoquery di una clausola WHERE; non è possibile definire una CTE ricorsiva nella sottoquery. La CTE ricorsiva deve precedere la clausola UPDATE.)
- DELETE

Se la clausola FROM di una query che contiene una clausola WITH non fa riferimento a nessuna delle tabelle definite dalla clausola WITH, la clausola WITH viene ignorata e la query viene eseguita normalmente.

A una tabella definita da una sottoquery della clausola WITH è possibile fare riferimento solo nell'ambito della query SELECT avviata dalla clausola WITH. Ad esempio, puoi fare riferimento a una tabella di questo tipo nella clausola FROM di una sottoquery nell'elenco SELECT, nella clausola WHERE o nella clausola HAVING. Non puoi utilizzare una clausola WITH in una sottoquery e fare riferimento alla tabella nella clausola FROM della query principale o un'altra sottoquery. Questo modello di query genera un messaggio di errore nel formato `relation table_name doesn't exist` per la tabella della clausola WITH.

Non puoi specificare un'altra clausola WITH all'interno di una sottoquery della clausola WITH.

Non puoi creare riferimenti alle tabelle definite dalle sottoquery della clausola WITH. Ad esempio, la seguente query restituisce un errore a causa del riferimento in avanti alla tabella W2 nella definizione della tabella W1:

```
with w1 as (select * from w2), w2 as (select * from w1)
select * from sales;
ERROR:  relation "w2" does not exist
```

Una sottoquery della clausola WITH non può consistere in un'istruzione SELECT INTO; tuttavia, è possibile utilizzare una clausola WITH in un'istruzione SELECT INTO.

Espressioni di tabella comuni ricorsive

Una espressione di tabella comune (CTE) ricorsiva è una CTE che fa riferimento a sé stessa. Una CTE ricorsiva è utile per eseguire query su dati gerarchici, ad esempio organigrammi che mostrano

relazioni di reporting tra dipendenti e responsabili. Per informazioni, consultare Esempio: CTE ricorsiva.

Un altro uso comune è una distinta base multilivello, quando un prodotto è costituito da molti componenti e ogni componente è costituito a sua volta da altri componenti o sottoinsiemi.

Assicurarsi di limitare la profondità di ricorsione includendo una clausola WHERE nella seconda sottoquery SELECT della query ricorsiva. Per un esempio, consultare Esempio: CTE ricorsiva. In caso contrario, può verificarsi un errore simile a quello riportato di seguito.

- `Recursive CTE out of working buffers.`
- `Exceeded recursive CTE max rows limit, please add correct CTE termination predicates or change the max_recursion_rows parameter.`

> **Note**
>
> `max_recursion_rows` è un parametro che imposta il numero massimo di righe che un CTE ricorsivo può restituire per evitare cicli di ricorsione infiniti. Si consiglia di non modificarlo con un valore superiore a quello predefinito. In questo modo si evita che infiniti problemi di ricorsione nelle query occupino uno spazio eccessivo nel cluster.

È possibile specificare un ordinamento e limitare il risultato della CTE ricorsiva. È possibile includere opzioni di raggruppamento e distinte sul risultato finale della CTE ricorsiva.

Non è possibile specificare un'altra clausola WITH RECURSIVE all'interno di una sottoquery. Il membro recursive_query non può includere una clausola order by o limit.

Esempi

L'esempio seguente mostra il caso più semplice possibile di una query che contiene una clausola WITH. La query WITH denominata VENUECOPY seleziona tutte le righe dalla tabella VENUE. La query principale a sua volta seleziona tutte le righe da VENUECOPY. La tabella VENUECOPY esiste solo per la durata di questa query.

```
with venuecopy as (select * from venue)
select * from venuecopy order by 1 limit 10;
```

```
 venueid |         venuename          |    venuecity    | venuestate | venueseats
```

```
---------+---------------------------+-----------------+------------+------------
1 | Toyota Park                | Bridgeview      | IL         |          0
2 | Columbus Crew Stadium      | Columbus        | OH         |          0
3 | RFK Stadium                | Washington      | DC         |          0
4 | CommunityAmerica Ballpark  | Kansas City     | KS         |          0
5 | Gillette Stadium           | Foxborough      | MA         |      68756
6 | New York Giants Stadium    | East Rutherford | NJ         |      80242
7 | BMO Field                  | Toronto         | ON         |          0
8 | The Home Depot Center      | Carson          | CA         |          0
9 | Dick's Sporting Goods Park | Commerce City   | CO         |          0
v     10 | Pizza Hut Park             | Frisco          | TX         |          0
(10 rows)
```

L'esempio seguente mostra una clausola WITH che produce due tabelle, denominate VENUE_SALES e TOP_VENUES. La seconda tabella della query WITH seleziona dalla prima. A sua volta, la clausola WHERE del blocco di query principale contiene una sottoquery che vincola la tabella TOP_VENUES.

```
with venue_sales as
(select venuename, venuecity, sum(pricepaid) as venuename_sales
from sales, venue, event
where venue.venueid=event.venueid and event.eventid=sales.eventid
group by venuename, venuecity),

top_venues as
(select venuename
from venue_sales
where venuename_sales > 800000)

select venuename, venuecity, venuestate,
sum(qtysold) as venue_qty,
sum(pricepaid) as venue_sales
from sales, venue, event
where venue.venueid=event.venueid and event.eventid=sales.eventid
and venuename in(select venuename from top_venues)
group by venuename, venuecity, venuestate
order by venuename;
```

```
        venuename       |   venuecity   | venuestate | venue_qty | venue_sales
------------------------+---------------+------------+-----------+-------------
August Wilson Theatre   | New York City | NY         |      3187 |  1032156.00
Biltmore Theatre        | New York City | NY         |      2629 |   828981.00
```

```
Charles Playhouse       | Boston        | MA         |      2502 |   857031.00
Ethel Barrymore Theatre | New York City | NY         |      2828 |   891172.00
Eugene O'Neill Theatre  | New York City | NY         |      2488 |   828950.00
Greek Theatre           | Los Angeles   | CA         |      2445 |   838918.00
Helen Hayes Theatre     | New York City | NY         |      2948 |   978765.00
Hilton Theatre          | New York City | NY         |      2999 |   885686.00
Imperial Theatre        | New York City | NY         |      2702 |   877993.00
Lunt-Fontanne Theatre   | New York City | NY         |      3326 |  1115182.00
Majestic Theatre        | New York City | NY         |      2549 |   894275.00
Nederlander Theatre     | New York City | NY         |      2934 |   936312.00
Pasadena Playhouse      | Pasadena      | CA         |      2739 |   820435.00
Winter Garden Theatre   | New York City | NY         |      2838 |   939257.00
(14 rows)
```

I seguenti due esempi illustrano le regole per l'ambito dei riferimenti di tabella basati sulle sottoquery della clausola WITH. La prima query viene eseguita, ma la seconda non riesce con un errore previsto. La prima query contiene la sottoquery clausola WITH all'interno dell'elenco SELECT della query principale. Alla tabella definita dalla clausola WITH (HOLIDAYS) si fa riferimento nella clausola FROM della sottoquery nell'elenco SELECT:

```
select caldate, sum(pricepaid) as daysales,
(with holidays as (select * from date where holiday ='t')
select sum(pricepaid)
from sales join holidays on sales.dateid=holidays.dateid
where caldate='2008-12-25') as dec25sales
from sales join date on sales.dateid=date.dateid
where caldate in('2008-12-25','2008-12-31')
group by caldate
order by caldate;

caldate   | daysales | dec25sales
----------+----------+------------
2008-12-25 | 70402.00 |   70402.00
2008-12-31 | 12678.00 |   70402.00
(2 rows)
```

La seconda query non riesce perché tenta di fare riferimento alla tabella HOLIDAYS nella query principale e nella sottoquery elenco SELECT. I riferimenti della query principale sono fuori ambito.

```
select caldate, sum(pricepaid) as daysales,
(with holidays as (select * from date where holiday ='t')
select sum(pricepaid)
```

```
from sales join holidays on sales.dateid=holidays.dateid
where caldate='2008-12-25') as dec25sales
from sales join holidays on sales.dateid=holidays.dateid
where caldate in('2008-12-25','2008-12-31')
group by caldate
order by caldate;

ERROR:  relation "holidays" does not exist
```

Esempio: CTE ricorsiva

Di seguito è riportato un esempio di CTE ricorsiva che restituisce i dipendenti che riportano direttamente o indirettamente a John. La query ricorsiva contiene una clausola WHERE per limitare la profondità di ricorsione a meno di 4 livelli.

```
--create and populate the sample table
  create table employee (
  id int,
  name varchar (20),
  manager_id int
  );

  insert into employee(id, name, manager_id)  values
(100, 'Carlos', null),
(101, 'John', 100),
(102, 'Jorge', 101),
(103, 'Kwaku', 101),
(110, 'Liu', 101),
(106, 'Mateo', 102),
(110, 'Nikki', 103),
(104, 'Paulo', 103),
(105, 'Richard', 103),
(120, 'Saanvi', 104),
(200, 'Shirley', 104),
(201, 'Sofía', 102),
(205, 'Zhang', 104);

--run the recursive query
  with recursive john_org(id, name, manager_id, level) as
( select id, name, manager_id, 1 as level
  from employee
  where name = 'John'
  union all
```

```
  select e.id, e.name, e.manager_id, level + 1 as next_level
  from employee e, john_org j
  where e.manager_id = j.id and level < 4
  )
select distinct id, name, manager_id from john_org order by manager_id;
```

Di seguito è riportato il risultato della query.

```
  id        name      manager_id
------+-----------+--------------
 101    John           100
 102    Jorge          101
 103    Kwaku          101
 110    Liu            101
 201    Sofía          102
 106    Mateo          102
 110    Nikki          103
 104    Paulo          103
 105    Richard        103
 120    Saanvi         104
 200    Shirley        104
 205    Zhang          104
```

Di seguito è riportato un organigramma per il dipartimento di John.

Elenco SELECT

Argomenti

- Sintassi
- Parametri
- Note per l'utilizzo
- Esempi

L'elenco SELECT assegna un nome a colonne, funzioni ed espressioni che la query deve restituire. L'elenco rappresenta l'output della query.

Per ulteriori informazioni sulle funzioni SQL, consulta Informazioni di riferimento sulle funzioni SQL. Per ulteriori informazioni sulle espressioni, consulta Espressioni condizionali.

Sintassi

```
SELECT
[ TOP number ]
[ ALL | DISTINCT ] * | expression [ AS column_alias ] [, ...]
```

Parametri

TOP number

TOP accetta come argomento un integer positivo che definisce il numero di righe che vengono restituite al client. Il comportamento con la clausola TOP è uguale al comportamento con la clausola LIMIT. A differenza del set di righe, il numero di righe restituito è fisso. Per restituire un set coerente di righe, utilizza TOP o LIMIT insieme a una clausola ORDER BY.

ALL

Parola chiave ridondante che definisce il comportamento predefinito se non specifichi DISTINCT. `SELECT ALL *` è identico a `SELECT *` (seleziona tutte le righe per tutte le colonne e conserva i duplicati).

DISTINCT

Opzione che elimina le righe duplicate dal set di risultati, in base ai valori corrispondenti in una o più colonne.

> **Note**
>
> Se l'applicazione consente chiavi esterne o primarie non valide, le query possono restituire risultati errati. Ad esempio, una query SELECT DISTINCT potrebbe restituire righe duplicate se la colonna della chiave primaria non contiene tutti i valori univoci. Per ulteriori informazioni, consulta Definizione di limitazioni delle tabelle.

* (asterisco)

Restituisce l'intero contenuto della tabella (tutte le colonne e tutte le righe).

espressione

Espressione formata da una o più colonne presenti nelle tabelle a cui fa riferimento la query. Un'espressione può contenere funzioni SQL. Ad esempio:

```
avg(datediff(day, listtime, saletime))
```

AS column_alias

Nome temporaneo per la colonna che viene utilizzata nel set di risultati finale. La parola chiave AS è facoltativa. Ad esempio:

```
avg(datediff(day, listtime, saletime)) as avgwait
```

Se non specifichi un alias per un'espressione che non è un semplice nome di colonna, il set di risultati applica un nome predefinito a quella colonna.

> **Note**
>
> L'alias viene riconosciuto subito dopo essere stato definito nell'elenco di destinazione. Puoi utilizzare un alias in altre espressioni definite successivamente nello stesso elenco di destinazione. Nell'esempio seguente viene descritto quanto segue.
>
> ```
> select clicks / impressions as probability, round(100 * probability, 1) as
>  percentage from raw_data;
> ```
>
> Il vantaggio del riferimento alias laterale è che non devi ripetere l'espressione con alias quando crei espressioni più complesse nello stesso elenco di destinazione. Quando Amazon Redshift analizza questo tipo di riferimento, gli alias definiti precedentemente vengono posti in linea. Se esiste una colonna con lo stesso nome definita nella clausola `FROM` come espressione con alias precedente, la colonna nella clausola `FROM` ha la priorità. Ad esempio, nella query precedente se è presente una colonna denominata "probability" nella tabella raw_data, la "probability" nella seconda espressione nell'elenco di destinazione fa riferimento a tale colonna invece che al nome alias "probability".

Note per l'utilizzo

TOP è un'estensione SQL e costituisce un'alternativa al comportamento di LIMIT. Non puoi utilizzare TOP e LIMIT nella stessa query.

Esempi

L'esempio seguente restituisce 10 righe dalla tabella SALES. Sebbene la query utilizzi la clausola TOP, restituisce comunque un set di righe non prevedibile perché non è specificata alcuna clausola ORDER BY:

```
select top 10 *
from sales;
```

La seguente query è funzionalmente equivalente, ma utilizza una clausola LIMIT invece di una clausola TOP:

```
select *
from sales
limit 10;
```

Il seguente esempio restituisce le prime 10 righe dalla tabella SALES mediante la clausola TOP, ordinate in base alla colonna QTYSOLD in ordine decrescente.

```
select top 10 qtysold, sellerid
from sales
order by qtysold desc, sellerid;

qtysold | sellerid
--------+----------
8 |      518
8 |      520
8 |      574
8 |      718
8 |      868
8 |     2663
8 |     3396
8 |     3726
8 |     5250
8 |     6216
(10 rows)
```

Il seguente esempio restituisce i primi due valori QTYSOLD e SELLERID dalla tabella SALES, ordinati in base alla colonna QTYSOLD:

```
select top 2 qtysold, sellerid
```

```
from sales
order by qtysold desc, sellerid;

qtysold | sellerid
--------+----------
8 |      518
8 |      520
(2 rows)
```

L'esempio seguente mostra l'elenco di gruppi di categorie distinti dalla tabella CATEGORY:

```
select distinct catgroup from category
order by 1;

catgroup
----------
Concerts
Shows
Sports
(3 rows)

--the same query, run without distinct
select catgroup from category
order by 1;

catgroup
----------
Concerts
Concerts
Concerts
Shows
Shows
Shows
Sports
Sports
Sports
Sports
Sports
(11 rows)
```

L'esempio seguente restituisce il set distinto di numeri delle settimane per dicembre 2008. Senza la clausola DISTINCT, l'istruzione restituirebbe 31 righe o una per ogni giorno del mese.

```
select distinct week, month, year
from date
where month='DEC' and year=2008
order by 1, 2, 3;

week | month | year
-----+-------+------
49 | DEC   | 2008
50 | DEC   | 2008
51 | DEC   | 2008
52 | DEC   | 2008
53 | DEC   | 2008
(5 rows)
```

Clausola FROM

La clausola FROM in una query elenca i riferimenti di tabella (tabelle, viste e sottoquery) da cui vengono selezionati i dati. Se sono elencati più riferimenti tabella, è necessario unire le tabelle, utilizzando la sintassi appropriata nella clausola FROM o nella clausola WHERE. Se non vengono specificati criteri di join, il sistema elabora la query come cross-join (prodotto cartesiano).

Argomenti

Sintassi

```
FROM table_reference [, ...]
```

dove table_reference è una delle opzioni seguenti:

```
with_subquery_table_name [ table_alias ]
table_name [ * ] [ table_alias ]
```

```
( subquery ) [ table_alias ]
table_reference [ NATURAL ] join_type table_reference
   [ ON join_condition | USING ( join_column [, ...] ) ]
table_reference PIVOT (
   aggregate(expr) [ [ AS ] aggregate_alias ]
   FOR column_name IN ( expression [ AS ] in_alias [, ...] )
) [ table_alias ]
table_reference UNPIVOT [ INCLUDE NULLS | EXCLUDE NULLS ] (
   value_column_name
   FOR name_column_name IN ( column_reference [ [ AS ]
   in_alias ] [, ...] )
) [ table_alias ]
UNPIVOT expression AS value_alias [ AT attribute_alias ]
```

L'opzione table_alias può essere utilizzato per assegnare nomi temporanei a tabelle e riferimenti a tabelle complessi e, se lo si desidera, anche alle relative colonne, come segue:

```
[ AS ] alias [ ( column_alias [, ...] ) ]
```

Parametri

with_subquery_table_name

Tabella definita da una sottoquery nella Clausola WITH.

table_name

Nome di una tabella o vista.

alias

Nome alternativo temporaneo per una tabella o vista. L'alias è obbligatorio per una tabella derivata da una sottoquery. In altri riferimenti di tabella, gli alias sono facoltativi. La parola chiave AS è sempre facoltativa. Gli alias di tabella forniscono una comoda scelta rapida per identificare le tabelle in altre parti di una query, come nella clausola WHERE. Ad esempio:

```
select * from sales s, listing l
where s.listid=l.listid
```

column_alias

Un'espressione semplice che restituisce un valore.

subquery

Espressione della query che restituisce una tabella. La tabella esiste solo per la durata della query e in genere le viene assegnato un nome o un alias; tuttavia l'alias non è obbligatorio. Puoi anche definire i nomi delle colonne per le tabelle che derivano da sottoquery. L'assegnazione degli alias alle colonne è importante quando vuoi unire i risultati delle sottoquery ad altre tabelle e quando vuoi selezionare o vincolare tali colonne altrove nella query.

Una sottoquery può contenere una clausola ORDER BY, ma questa potrebbe non avere alcun effetto se non viene specificata una clausola LIMIT o OFFSET.

NATURAL

Definisce un join che utilizza automaticamente tutte le coppie di colonne con lo stesso nome nelle due tabelle come colonne di unione. La condizione di join esplicita non è obbligatoria. Ad esempio, se le tabelle CATEGORY ed EVENT hanno entrambe le colonne denominate CATID, un join naturale di tali tabelle è un join sulle rispettive colonne CATID.

> **Note**
>
> Se viene specificato un join NATURAL ma non esistono coppie di colonne con nome identico nelle tabelle da unire, la query viene impostata automaticamente su un cross-join.

join_type

Specifica uno dei seguenti tipi di join:

- [INNER] JOIN
- LEFT [OUTER] JOIN
- RIGHT [OUTER] JOIN
- FULL [OUTER] JOIN
- CROSS JOIN

I cross-join sono join non qualificati; restituiscono il prodotto cartesiano delle due tabelle.

I join inner e outer sono join qualificati. Sono qualificati in modo implicito (in join naturali) con la sintassi ON o USING nella clausola FROM o con una condizione della clausola WHERE.

Un inner join restituisce solo le righe corrispondenti, in base alla condizione di join o all'elenco delle colonne di join. Un outer join restituisce tutte le righe che l'inner join equivalente

restituirebbe, più le righe non corrispondenti dalla tabella "left", dalla tabella "right" o da entrambe le tabelle. La tabella di sinistra è la tabella elencata per prima e la tabella di destra è la seconda tabella nell'elenco. Le righe non corrispondenti contengono valori NULL per riempire gli spazi vuoti nelle colonne di output.

ON join_condition

Tipo di specifica del join in cui le colonne di unione vengono dichiarate come una condizione che segue la parola chiave ON. Ad esempio:

```
sales join listing
on sales.listid=listing.listid and sales.eventid=listing.eventid
```

USING (join_column [, ...])

Tipo di specifica del join in cui le colonne di unione vengono elencate tra parentesi. Se vengono specificate più colonne di unione, queste sono delimitate da virgole. La parola chiave USING deve precedere l'elenco. Per esempio:

```
sales join listing
using (listid,eventid)
```

PIVOT

Ruota l'output da righe a colonne, allo scopo di rappresentare i dati tabulari in un formato di facile lettura. L'output è rappresentato orizzontalmente su più colonne. PIVOT è simile a una query GROUP BY con un'aggregazione, utilizzando un'espressione aggregata per specificare un formato di output. Tuttavia, a differenza di GROUP BY, i risultati vengono restituiti in colonne anziché in righe.

Per gli esempi di query con PIVOT e UNPIVOT consulta Esempi PIVOT e UNPIVOT.

UNPIVOT

Rotazione delle colonne in righe con UNPIVOT: l'operatore trasforma le colonne dei risultati, da una tabella di input o dai risultati di una query, in righe, per rendere l'output più facile da leggere. UNPIVOT combina i dati delle colonne di input in due colonne di risultato: una colonna del nome e una colonna di valore. La colonna del nome contiene i nomi delle colonne dell'input, come voci di riga. La colonna dei valori contiene i valori delle colonne di input, ad esempio i risultati di un'aggregazione. Ad esempio, il numero di articoli in varie categorie.

Scompattamento degli oggetti con UNPIVOT (SUPER): è possibile eseguire l'unpivot degli oggetti, dove espressione è un'espressione SUPER che si riferisce a un altro elemento della clausola FROM. Per ulteriori informazioni, consulta Nidificazione di oggetti. Contiene anche esempi che mostrano come interrogare dati semistrutturati, ad esempio dati in formato JSON.

Note per l'utilizzo

Le colonne di unione devono avere tipi di dati comparabili.

Un join NATURAL o USING conserva solo una di ciascuna coppia di colonne di unione nel set di risultati intermedi.

Un join con la sintassi ON mantiene entrambe le colonne di unione nel set di risultati intermedi.

consultare anche Clausola WITH.

Esempi PIVOT e UNPIVOT

PIVOT e UNPIVOT sono parametri della clausola FROM che ruotano rispettivamente l'output della query da righe a colonne e da colonne a righe. Rappresentano i risultati delle query tabulari in un formato facile da leggere. Gli esempi seguenti utilizzano dati e query di prova per mostrare come utilizzarli.

Per ulteriori informazioni su questi e altri parametri, consulta Clausola FROM.

Esempi PIVOT

Impostare la tabella di esempio e i dati e utilizzarli per eseguire le query di esempio successive.

```
CREATE TABLE part (
    partname varchar,
    manufacturer varchar,
    quality int,
    price decimal(12, 2)
);

INSERT INTO part VALUES ('prop', 'local parts co', 2, 10.00);
INSERT INTO part VALUES ('prop', 'big parts co', NULL, 9.00);
INSERT INTO part VALUES ('prop', 'small parts co', 1, 12.00);

INSERT INTO part VALUES ('rudder', 'local parts co', 1, 2.50);
INSERT INTO part VALUES ('rudder', 'big parts co', 2, 3.75);
INSERT INTO part VALUES ('rudder', 'small parts co', NULL, 1.90);
```

```
INSERT INTO part VALUES ('wing', 'local parts co', NULL, 7.50);
INSERT INTO part VALUES ('wing', 'big parts co', 1, 15.20);
INSERT INTO part VALUES ('wing', 'small parts co', NULL, 11.80);
```

PIVOT su `partname` con un'aggregazione `AVG` su`price`.

```
SELECT *
FROM (SELECT partname, price FROM part) PIVOT (
    AVG(price) FOR partname IN ('prop', 'rudder', 'wing')
);
```

La query restituisce i seguenti risultati:

```
  prop   |  rudder  |  wing
---------+----------+---------
 10.33   | 2.71     | 11.50
```

Nell'esempio precedente, i risultati vengono trasformati in colonne. L'esempio seguente mostra una query `GROUP BY` che restituisce i prezzi medi in righe anziché in colonne.

```
SELECT partname, avg(price)
FROM (SELECT partname, price FROM part)
WHERE partname IN ('prop', 'rudder', 'wing')
GROUP BY partname;
```

La query restituisce i seguenti risultati:

```
 partname |  avg
----------+-------
 prop     | 10.33
 rudder   |  2.71
 wing     | 11.50
```

Un esempio `PIVOT` con `manufacturer` come colonna implicita.

```
SELECT *
FROM (SELECT quality, manufacturer FROM part) PIVOT (
    count(*) FOR quality IN (1, 2, NULL)
);
```

La query restituisce i seguenti risultati:

```
 manufacturer      | 1  | 2  | null
-------------------+----+----+------
 local parts co    | 1  | 1  |  1
 big parts co      | 1  | 1  |  1
 small parts co    | 1  | 0  |  2
```

Le colonne della tabella di input che non sono referenziate nella definizione PIVOT sono aggiunte implicitamente alla tabella dei risultati. Questo è il caso della colonna manufacturer dell'esempio precedente. L'esempio dimostra anche che NULL è un valore valido per l'operatore IN.

PIVOT nell'esempio precedente restituisce informazioni simili a quelle della seguente query, che include GROUP BY. La differenza è che PIVOT restituisce il valore 0 per colonna 2 e il produttore small parts co. La query GROUP BY non contiene una riga corrispondente. Nella maggior parte dei casi PIVOT inserisce NULL se una riga non contiene dati di input per una determinata colonna. Tuttavia, l'aggregato di conteggio non restituisce NULL e 0 viene utilizzato come valore predefinito.

```
SELECT manufacturer, quality, count(*)
FROM (SELECT quality, manufacturer FROM part)
WHERE quality IN (1, 2) OR quality IS NULL
GROUP BY manufacturer, quality
ORDER BY manufacturer;
```

La query restituisce i seguenti risultati:

```
 manufacturer        | quality | count
---------------------+---------+-------
 big parts co        |         |     1
 big parts co        |       2 |     1
 big parts co        |       1 |     1
 local parts co      |       2 |     1
 local parts co      |       1 |     1
 local parts co      |         |     1
 small parts co      |       1 |     1
 small parts co      |         |     2
```

L'operatore PIVOT accetta alias opzionali sull'espressione aggregata e su ciascun valore per l'operatore IN. Usa gli alias per personalizzare i nomi delle colonne. Se non esiste un alias aggregato, solo gli alias dell'elenco IN. In caso contrario, l'alias aggregato viene aggiunto al nome della colonna con un trattino di sottolineatura per separare i nomi.

```
SELECT *
FROM (SELECT quality, manufacturer FROM part) PIVOT (
    count(*) AS count FOR quality IN (1 AS high, 2 AS low, NULL AS na)
);
```

La query restituisce i seguenti risultati:

```
 manufacturer      | high_count  | low_count | na_count
-------------------+-------------+-----------+----------
 local parts co    |           1 |         1 |        1
 big parts co      |           1 |         1 |        1
 small parts co    |           1 |         0 |        2
```

Imposta la tabella e i dati di esempio seguenti e utilizzali per eseguire le query di esempio successive. I dati rappresentano le date di prenotazione di una raccolta di hotel.

```
CREATE TABLE bookings (
    booking_id int,
    hotel_code char(8),
    booking_date date,
    price decimal(12, 2)
);

INSERT INTO bookings VALUES (1, 'FOREST_L', '02/01/2023', 75.12);
INSERT INTO bookings VALUES (2, 'FOREST_L', '02/02/2023', 75.00);
INSERT INTO bookings VALUES (3, 'FOREST_L', '02/04/2023', 85.54);

INSERT INTO bookings VALUES (4, 'FOREST_L', '02/08/2023', 75.00);
INSERT INTO bookings VALUES (5, 'FOREST_L', '02/11/2023', 75.00);
INSERT INTO bookings VALUES (6, 'FOREST_L', '02/14/2023', 90.00);

INSERT INTO bookings VALUES (7, 'FOREST_L', '02/21/2023', 60.00);
INSERT INTO bookings VALUES (8, 'FOREST_L', '02/22/2023', 85.00);
INSERT INTO bookings VALUES (9, 'FOREST_L', '02/27/2023', 90.00);

INSERT INTO bookings VALUES (10, 'DESERT_S', '02/01/2023', 98.00);
INSERT INTO bookings VALUES (11, 'DESERT_S', '02/02/2023', 75.00);
INSERT INTO bookings VALUES (12, 'DESERT_S', '02/04/2023', 85.00);

INSERT INTO bookings VALUES (13, 'DESERT_S', '02/05/2023', 75.00);
INSERT INTO bookings VALUES (14, 'DESERT_S', '02/06/2023', 34.00);
INSERT INTO bookings VALUES (15, 'DESERT_S', '02/09/2023', 85.00);
```

```
INSERT INTO bookings VALUES (16, 'DESERT_S', '02/12/2023', 23.00);
INSERT INTO bookings VALUES (17, 'DESERT_S', '02/13/2023', 76.00);
INSERT INTO bookings VALUES (18, 'DESERT_S', '02/14/2023', 85.00);

INSERT INTO bookings VALUES (19, 'OCEAN_WV', '02/01/2023', 98.00);
INSERT INTO bookings VALUES (20, 'OCEAN_WV', '02/02/2023', 75.00);
INSERT INTO bookings VALUES (21, 'OCEAN_WV', '02/04/2023', 85.00);

INSERT INTO bookings VALUES (22, 'OCEAN_WV', '02/06/2023', 75.00);
INSERT INTO bookings VALUES (23, 'OCEAN_WV', '02/09/2023', 34.00);
INSERT INTO bookings VALUES (24, 'OCEAN_WV', '02/12/2023', 85.00);

INSERT INTO bookings VALUES (25, 'OCEAN_WV', '02/13/2023', 23.00);
INSERT INTO bookings VALUES (26, 'OCEAN_WV', '02/14/2023', 76.00);
INSERT INTO bookings VALUES (27, 'OCEAN_WV', '02/16/2023', 85.00);

INSERT INTO bookings VALUES (28, 'CITY_BLD', '02/01/2023', 98.00);
INSERT INTO bookings VALUES (29, 'CITY_BLD', '02/02/2023', 75.00);
INSERT INTO bookings VALUES (30, 'CITY_BLD', '02/04/2023', 85.00);

INSERT INTO bookings VALUES (31, 'CITY_BLD', '02/12/2023', 75.00);
INSERT INTO bookings VALUES (32, 'CITY_BLD', '02/13/2023', 34.00);
INSERT INTO bookings VALUES (33, 'CITY_BLD', '02/17/2023', 85.00);

INSERT INTO bookings VALUES (34, 'CITY_BLD', '02/22/2023', 23.00);
INSERT INTO bookings VALUES (35, 'CITY_BLD', '02/23/2023', 76.00);
INSERT INTO bookings VALUES (36, 'CITY_BLD', '02/24/2023', 85.00);
```

In questa query di esempio, i record delle prenotazioni vengono conteggiati per fornire il totale ogni settimana. La data di fine di ogni settimana diventa un nome di colonna.

```
SELECT * FROM
    (SELECT
       booking_id,
       (date_trunc('week', booking_date::date) + '5 days'::interval)::date as enddate,
       hotel_code AS "hotel code"
FROM bookings
) PIVOT (
    count(booking_id) FOR enddate IN ('2023-02-04','2023-02-11','2023-02-18')
);
```

La query restituisce i seguenti risultati:

```
 hotel code | 2023-02-04  | 2023-02-11 | 2023-02-18
------------+-------------+------------+----------
 FOREST_L   |           3 |          2 |        1
 DESERT_S   |           4 |          3 |        2
 OCEAN_WV   |           3 |          3 |        3
 CITY_BLD   |           3 |          1 |        2
```

Amazon Redshift non supporta l'uso di CROSSTAB per il passaggio in più colonne. È tuttavia possibile modificare i dati delle righe nelle colonne, in modo simile a un'aggregazione con PIVOT, utilizzando una query simile alla seguente. Usa gli stessi dati delle prenotazioni dell'esempio precedente.

```
SELECT
  booking_date,
  MAX(CASE WHEN hotel_code = 'FOREST_L' THEN 'forest is booked' ELSE '' END) AS
 FOREST_L,
  MAX(CASE WHEN hotel_code = 'DESERT_S' THEN 'desert is booked' ELSE '' END) AS
 DESERT_S,
  MAX(CASE WHEN hotel_code = 'OCEAN_WV' THEN 'ocean is booked' ELSE '' END)  AS
 OCEAN_WV
FROM bookings
GROUP BY booking_date
ORDER BY booking_date asc;
```

La query di esempio restituisce le date di prenotazione elencate accanto a una breve frase che indica quali hotel sono prenotati.

```
 booking_date  | forest_l         | desert_s         | ocean_wv
---------------+------------------+------------------+--------------------
 2023-02-01    | forest is booked | desert is booked |  ocean is booked
 2023-02-02    | forest is booked | desert is booked |  ocean is booked
 2023-02-04    | forest is booked | desert is booked |  ocean is booked
 2023-02-05    |                  | desert is booked |
 2023-02-06    |                  | desert is booked |
```

Di seguito sono riportate le note di utilizzo per `PIVOT`:

- `PIVOT` può essere applicato a tabelle, sottoquery ed espressioni di tabella comuni (CTE). `PIVOT` non può essere applicato ad alcuna espressione `JOIN`, CTE ricorrenti, `PIVOT`, oppure espressioni `UNPIVOT`. Inoltre non sono supportate le espressioni non annidate `SUPER` e le tabelle annidate di Redshift Spectrum.

- PIVOT supporta COUNT, SUM, MIN, MAX e le funzioni aggregate AVG.
- L'espressione aggregata PIVOT deve essere una chiamata di una funzione di aggregazione supportata. Le espressioni complesse sopra l'aggregato non sono supportate. Gli argomenti aggregati non possono contenere riferimenti a tabelle diverse dalla tabella di input PIVOT. Anche i riferimenti correlati a una query principale non sono supportati. L'argomento aggregato può contenere sottoquery. Questi possono essere correlati internamente o sulla tabella di input PIVOT.
- I valori della lista PIVOT IN non possono essere riferimenti di colonna o sottoquery. Ogni valore deve essere compatibile con la colonna di riferimento FOR.
- Se i valori della lista IN non hanno alias, PIVOT genera nomi di colonne predefiniti. Per i valori IN costanti come 'abc' o 5 il nome di colonna predefinito è la costante stessa. Per qualsiasi espressione complessa, il nome della colonna è un nome predefinito Amazon Redshift standard, ad esempio ?column?.

Esempi di UNPIVOT

Impostare i dati di esempio e utilizzarli per eseguire le query di esempio successive.

```
CREATE TABLE count_by_color (quality varchar, red int, green int, blue int);

INSERT INTO count_by_color VALUES ('high', 15, 20, 7);
INSERT INTO count_by_color VALUES ('normal', 35, NULL, 40);
INSERT INTO count_by_color VALUES ('low', 10, 23, NULL);
```

UNPIVOT sulle colonne di input rosso, verde e blu.

```
SELECT *
FROM (SELECT red, green, blue FROM count_by_color) UNPIVOT (
    cnt FOR color IN (red, green, blue)
);
```

La query restituisce i seguenti risultati:

```
 color | cnt
-------+-----
 red   |  15
 red   |  35
 red   |  10
 green |  20
```

```
 green |  23
 blue  |   7
 blue  |  40
```

Per impostazione predefinita,i valori NULL nella colonna di input vengono ignorati e non producono una riga di risultato.

L'esempio seguente mostra UNPIVOT con INCLUDE NULLS.

```
SELECT *
FROM (
    SELECT red, green, blue
    FROM count_by_color
) UNPIVOT INCLUDE NULLS (
    cnt FOR color IN (red, green, blue)
);
```

Di seguito è riportato l'output risultante.

```
 color | cnt
-------+-----
 red   |  15
 red   |  35
 red   |  10
 green |  20
 green |
 green |  23
 blue  |   7
 blue  |  40
 blue  |
```

Se il parametro INCLUDING NULLS è impostato,i valori di input NULL generano righe dei risultati.

The following query shows UNPIVOT con quality come colonna implicita.

```
SELECT *
FROM count_by_color UNPIVOT (
    cnt FOR color IN (red, green, blue)
);
```

La query restituisce i seguenti risultati:

```
 quality | color | cnt
---------+-------+-----
 high    | red   |  15
 normal  | red   |  35
 low     | red   |  10
 high    | green |  20
 low     | green |  23
 high    | blue  |   7
 normal  | blue  |  40
```

Le colonne della tabella di input a cui non si fa riferimento nella definizione UNPIVOT vengono aggiunte implicitamente alla tabella dei risultati. Nell'esempio, questo è il caso della colonna `quality`.

L'esempio seguente mostra UNPIVOT con alias per i valori nell'elenco IN.

```
SELECT *
FROM count_by_color UNPIVOT (
    cnt FOR color IN (red AS r, green AS g, blue AS b)
);
```

La query precedente restituisce i seguenti risultati:

```
 quality | color | cnt
---------+-------+-----
 high    | r     |  15
 normal  | r     |  35
 low     | r     |  10
 high    | g     |  20
 low     | g     |  23
 high    | b     |   7
 normal  | b     |  40
```

L'operatore UNPIVOT accetta alias opzionali su ciascuno valore di elenco IN. Ogni alias fornisce la personalizzazione dei dati in ciascuna colonna `value`.

Di seguito sono riportate le note di utilizzo per UNPIVOT.

- UNPIVOT può essere applicato a tabelle, sottoquery ed espressioni di tabella comuni (CTE). UNPIVOT non può essere applicato ad alcuna espressione JOIN, CTE ricorrenti, PIVOT, oppure

espressioni UNPIVOT. Inoltre non sono supportate le espressioni non annidate SUPER e le tabelle annidate di Redshift Spectrum.

- L'elenco UNPIVOT IN deve contenere solo i riferimenti alle colonne della tabella di input. Le colonne dell'elenco IN devono avere un tipo comune con cui sono tutte compatibili. La colonna valore UNPIVOT ha questo tipo comune. La colonna del nome UNPIVOT è di tipo VARCHAR.
- Se un valore di elenco IN non ha un alias, UNPIVOT utilizza il nome della colonna come valore predefinito.

Esempi di JOIN

Una clausola SQL JOIN viene utilizzata per combinare i dati di due o più tabelle in base a campi comuni. I risultati potrebbero cambiare o meno a seconda del metodo di join specificato. Per ulteriori informazioni sulla sintassi di una clausola JOIN, consulta Parametri.

Gli esempi seguenti utilizzano i dati dai dati di esempio TICKIT. Per ulteriori informazioni sullo schema di database, consulta Database di esempio. Per informazioni su come caricare dati di esempio, consulta Loading data nella Amazon Redshift Getting Started Guide.

La seguente query è un inner join (senza la parola chiave JOIN) tra la tabella LISTING e la tabella SALES, in cui il LISTID della tabella LISTING è compreso tra 1 e 5. Questa query corrisponde ai valori della colonna LISTID nella tabella LISTING (la tabella di sinistra) e SALES (la tabella di destra). I risultati mostrano che LISTID 1, 4 e 5 corrispondono ai criteri.

```
select listing.listid, sum(pricepaid) as price, sum(commission) as comm
from listing, sales
where listing.listid = sales.listid
and listing.listid between 1 and 5
group by 1
order by 1;

listid | price  |  comm
-------+--------+--------
     1 | 728.00 | 109.20
     4 |  76.00 |  11.40
     5 | 525.00 |  78.75
```

La seguente query è un outer join. Gli outer join sinistro e destro mantengono i valori da una delle tabelle unite quando non viene trovata alcuna corrispondenza nell'altra tabella. Le tabelle sinistra e destra sono la prima e la seconda tabella elencate nella sintassi. I valori NULL vengono utilizzati per

riempire gli "spazi vuoti" nel set di risultati. Questa query corrisponde ai valori della colonna LISTID nella tabella LISTING (la tabella di sinistra) e SALES (la tabella di destra). I risultati mostrano che LISTID 2 e 3 non hanno comportato vendite.

```
select listing.listid, sum(pricepaid) as price, sum(commission) as comm
from listing left outer join sales on sales.listid = listing.listid
where listing.listid between 1 and 5
group by 1
order by 1;

listid | price  |  comm
-------+--------+--------
     1 | 728.00 | 109.20
     2 | NULL   | NULL
     3 | NULL   | NULL
     4 |  76.00 |  11.40
     5 | 525.00 |  78.75
```

La seguente query è un outer join. Questa query corrisponde ai valori della colonna LISTID nella tabella LISTING (la tabella di sinistra) e SALES (la tabella di destra). I risultati mostrano che LISTID 1, 4 e 5 corrispondono ai criteri.

```
select listing.listid, sum(pricepaid) as price, sum(commission) as comm
from listing right outer join sales on sales.listid = listing.listid
where listing.listid between 1 and 5
group by 1
order by 1;

listid | price  |  comm
-------+--------+--------
     1 | 728.00 | 109.20
     4 |  76.00 |  11.40
     5 | 525.00 |  78.75
```

La seguente query è un fullr join. I full join mantengono i valori da una delle tabelle unite quando non viene trovata alcuna corrispondenza nell'altra tabella. Le tabelle sinistra e destra sono la prima e la seconda tabella elencate nella sintassi. I valori NULL vengono utilizzati per riempire gli "spazi vuoti" nel set di risultati. Questa query corrisponde ai valori della colonna LISTID nella tabella LISTING (la tabella di sinistra) e SALES (la tabella di destra). I risultati mostrano che LISTID 2 e 3 non hanno comportato vendite.

```
select listing.listid, sum(pricepaid) as price, sum(commission) as comm
from listing full join sales on sales.listid = listing.listid
where listing.listid between 1 and 5
group by 1
order by 1;

listid | price  |  comm
-------+--------+--------
     1 | 728.00 | 109.20
     2 | NULL   | NULL
     3 | NULL   | NULL
     4 |  76.00 |  11.40
     5 | 525.00 |  78.75
```

La seguente query è un full join. Questa query corrisponde ai valori della colonna LISTID nella tabella LISTING (la tabella di sinistra) e SALES (la tabella di destra). Solo le righe che non determinano alcuna vendita (LISTID 2 e 3) sono presenti nei risultati.

```
select listing.listid, sum(pricepaid) as price, sum(commission) as comm
from listing full join sales on sales.listid = listing.listid
where listing.listid between 1 and 5
and (listing.listid IS NULL or sales.listid IS NULL)
group by 1
order by 1;

listid | price  |  comm
-------+--------+--------
     2 | NULL   | NULL
     3 | NULL   | NULL
```

Di seguito è riportato un esempio di inner join con la clausola ON. In questo caso, le righe NULL non vengono restituite.

```
select listing.listid, sum(pricepaid) as price, sum(commission) as comm
from sales join listing
on sales.listid=listing.listid and sales.eventid=listing.eventid
where listing.listid between 1 and 5
group by 1
order by 1;

listid | price  |  comm
-------+--------+--------
```

```
     1 | 728.00 | 109.20
     4 |  76.00 |  11.40
     5 | 525.00 |  78.75
```

La seguente query è un cross join o un join cartesiano della tabella LISTING e della tabella SALES con un predicato per limitare i risultati. Questa query corrisponde ai valori delle colonne LISTID nella tabella SALES e nella tabella LISTING per LISTID 1, 2, 3, 4 e 5 in entrambe le tabelle. I risultati mostrano che 20 righe soddisfano i criteri.

```
select sales.listid as sales_listid, listing.listid as listing_listid
from sales cross join listing
where sales.listid between 1 and 5
and listing.listid between 1 and 5
order by 1,2;

sales_listid | listing_listid
-------------+---------------
1            | 1
1            | 2
1            | 3
1            | 4
1            | 5
4            | 1
4            | 2
4            | 3
4            | 4
4            | 5
5            | 1
5            | 1
5            | 2
5            | 2
5            | 3
5            | 3
5            | 4
5            | 4
5            | 5
5            | 5
```

Di seguito è riportato un esempio di join naturale tra due tabelle. In questo caso, le colonne listid, sellerid, eventid e dateid hanno nomi e tipi di dati identici in entrambe le tabelle e quindi vengono utilizzate come colonne di join. I risultati sono limitati a cinque righe.

```
select listid, sellerid, eventid, dateid, numtickets
from listing natural join sales
order by 1
limit 5;

listid | sellerid  | eventid | dateid | numtickets
-------+-----------+---------+--------+-----------
113    | 29704     | 4699    | 2075   | 22
115    | 39115     | 3513    | 2062   | 14
116    | 43314     | 8675    | 1910   | 28
118    | 6079      | 1611    | 1862   | 9
163    | 24880     | 8253    | 1888   | 14
```

Di seguito è riportato un esempio di join tra due tabelle con la clausola USING. In questo caso, le colonne listid e eventid vengono utilizzate come colonne di join. I risultati sono limitati a cinque righe.

```
select listid, listing.sellerid, eventid, listing.dateid, numtickets
from listing join sales
using (listid, eventid)
order by 1
limit 5;

listid | sellerid | eventid | dateid | numtickets
-------+----------+---------+--------+-----------
1      | 36861    | 7872    | 1850   | 10
4      | 8117     | 4337    | 1970   | 8
5      | 1616     | 8647    | 1963   | 4
5      | 1616     | 8647    | 1963   | 4
6      | 47402    | 8240    | 2053   | 18
```

La seguente query è un inner join di due sottoquery della clausola FROM. La query trova il numero di biglietti venduti e invenduti per diverse categorie di eventi (concerti e spettacoli). Queste sottoquery della clausola FROM sono sottoquery table e possono restituire più colonne e righe.

```
select catgroup1, sold, unsold
from
(select catgroup, sum(qtysold) as sold
from category c, event e, sales s
where c.catid = e.catid and e.eventid = s.eventid
group by catgroup) as a(catgroup1, sold)
join
(select catgroup, sum(numtickets)-sum(qtysold) as unsold
```

```
from category c, event e, sales s, listing l
where c.catid = e.catid and e.eventid = s.eventid
and s.listid = l.listid
group by catgroup) as b(catgroup2, unsold)

on a.catgroup1 = b.catgroup2
order by 1;

catgroup1 |  sold  | unsold
----------+--------+--------
Concerts  | 195444 |1067199
Shows     | 149905 | 817736
```

Clausola WHERE

La clausola WHERE contiene le condizioni che possono unire tabelle o applicare predicati alle colonne nelle tabelle. Le tabelle possono inner join utilizzando la sintassi appropriata nella clausola WHERE o nella clausola FROM. I criteri degli outer join devono essere specificati nella clausola FROM.

Sintassi

```
[ WHERE condition ]
```

condizione

Qualsiasi condizione di ricerca con un risultato booleano, ad esempio una condizione di join o un predicato su una colonna della tabella. I seguenti esempi sono condizioni di join valide:

```
sales.listid=listing.listid
sales.listid<>listing.listid
```

I seguenti esempi sono condizioni valide sulle colonne delle tabelle:

```
catgroup like 'S%'
venueseats between 20000 and 50000
eventname in('Jersey Boys','Spamalot')
year=2008
length(catdesc)>25
date_part(month, caldate)=6
```

Le condizioni possono essere semplici o complesse; per le condizioni complesse, puoi utilizzare le parentesi per isolare le unità logiche. Nell'esempio seguente, la condizione di join è racchiusa tra parentesi.

```
where (category.catid=event.catid) and category.catid in(6,7,8)
```

Note per l'utilizzo

Puoi utilizzare gli alias nella clausola WHERE per fare riferimento alle espressioni di elenco selezionate.

Non puoi limitare i risultati delle funzioni di aggregazione nella clausola WHERE; utilizza la clausola HAVING per questo scopo.

Le colonne che sono limitate nella clausola WHERE devono derivare dai riferimenti di tabella nella clausola FROM.

Esempio

La seguente query utilizza una combinazione di diverse restrizioni della clausola WHERE, inclusa una condizione di join per le tabelle SALES ed EVENT, un predicato sulla colonna EVENTNAME e due predicati sulla colonna STARTTIME.

```
select eventname, starttime, pricepaid/qtysold as costperticket, qtysold
from sales, event
where sales.eventid = event.eventid
and eventname='Hannah Montana'
and date_part(quarter, starttime) in(1,2)
and date_part(year, starttime) = 2008
order by 3 desc, 4, 2, 1 limit 10;

eventname    |      starttime      |   costperticket   | qtysold
---------------+---------------------+-------------------+---------
Hannah Montana | 2008-06-07 14:00:00 |     1706.00000000 |       2
Hannah Montana | 2008-05-01 19:00:00 |     1658.00000000 |       2
Hannah Montana | 2008-06-07 14:00:00 |     1479.00000000 |       1
Hannah Montana | 2008-06-07 14:00:00 |     1479.00000000 |       3
Hannah Montana | 2008-06-07 14:00:00 |     1163.00000000 |       1
Hannah Montana | 2008-06-07 14:00:00 |     1163.00000000 |       2
Hannah Montana | 2008-06-07 14:00:00 |     1163.00000000 |       4
Hannah Montana | 2008-05-01 19:00:00 |      497.00000000 |       1
Hannah Montana | 2008-05-01 19:00:00 |      497.00000000 |       2
```

```
Hannah Montana | 2008-05-01 19:00:00 |      497.00000000 |       4
(10 rows)
```

Outer join di tipo Oracle nella clausola WHERE

Per la compatibilità con Oracle, Amazon Redshift supporta l'operatore outer join Oracle (+) nelle condizioni di join della clausola WHERE. Questo operatore è destinato all'uso solo nella definizione delle condizioni di outer join; non provare a usarlo in altri contesti. Altri usi di questo operatore sono silenziosamente ignorati nella maggior parte dei casi.

Un outer join restituisce tutte le righe che l'inner join equivalente restituirebbe, più le righe non corrispondenti da una o entrambe le tabelle. Nella clausola FROM, puoi specificare gli outer join sinistro, destro e completo. Nella clausola WHERE, puoi specificare solo gli outer join destro.

Per l'outer join delle tabelle TABLE1 e TABLE2 e per restituire le righe non corrispondenti da TABLE1 (un outer join sinistro), specifica `TABLE1 LEFT OUTER JOIN TABLE2` nella clausola FROM o applica l'operatore (+) a tutti i join colonne da TABLE2 nella clausola WHERE. Per tutte le righe in TABLE1 che non hanno righe corrispondenti in TABLE2, il risultato della query contiene valori null per qualsiasi espressione dell'elenco di selezione contenente colonne da TABLE2.

Per produrre lo stesso comportamento per tutte le righe in TABLE2 che non hanno righe corrispondenti in TABLE1, specifica `TABLE1 RIGHT OUTER JOIN TABLE2` nella clausola FROM o applica l'operatore (+) a tutti i join colonne da TABLE1 nella clausola WHERE.

Sintassi di base

```
[ WHERE {
[ table1.column1 = table2.column1(+) ]
[ table1.column1(+) = table2.column1 ]
}
```

La prima condizione è equivalente a:

```
from table1 left outer join table2
on table1.column1=table2.column1
```

La seconda condizione è equivalente a:

```
from table1 right outer join table2
on table1.column1=table2.column1
```

> **Note**
>
> La sintassi qui mostrata copre il caso semplice di un equijoin su una coppia di colonne di unione. Tuttavia, sono validi anche altri tipi di condizioni di confronto e più coppie di colonne di unione.

Ad esempio, la seguente clausola WHERE definisce un outer join su due coppie di colonne. L'operatore (+) deve essere collegato alla stessa tabella in entrambe le condizioni:

```
where table1.col1 > table2.col1(+)
and table1.col2 = table2.col2(+)
```

Note per l'utilizzo

Se possibile, utilizza la sintassi OUTER JOIN della clausola FROM standard anziché l'operatore (+) nella clausola WHERE. Le query che contengono l'operatore (+) sono soggette alle seguenti regole:

- Puoi utilizzare solo l'operatore (+) nella clausola WHERE e solo in riferimento alle colonne di tabelle o viste.
- Non puoi applicare l'operatore (+) alle espressioni. Tuttavia, un'espressione può contenere colonne che utilizzano l'operatore (+). Ad esempio, la seguente condizione di join restituisce un errore di sintassi:

  ```
  event.eventid*10(+)=category.catid
  ```

  Tuttavia, la seguente condizione di join è valida:

  ```
  event.eventid(+)*10=category.catid
  ```

- Non puoi utilizzare l'operatore (+) in un blocco di query che contiene anche la sintassi di join della clausola FROM.
- Se due tabelle sono unite in più condizioni di join, è necessario utilizzare l'operatore (+) in tutte o in nessuna di queste condizioni. Un join con stili di sintassi misti viene eseguito come inner join, senza avvertenza.
- L'operatore (+) non produce un outer join se si aggiunge una tabella nella query esterna con una tabella risultante da una query interna.

- Per utilizzare l'operatore (+) nell'outer join di una tabella su se stessa, è necessario definire gli alias di tabella nella clausola FROM e farvi riferimento nella condizione di join:

```
select count(*)
from event a, event b
where a.eventid(+)=b.catid;

count
-------
8798
(1 row)
```

- Non puoi combinare una condizione di join che contiene l'operatore (+) con una condizione OR o una condizione IN. Ad esempio:

```
select count(*) from sales, listing
where sales.listid(+)=listing.listid or sales.salesid=0;
ERROR:  Outer join operator (+) not allowed in operand of OR or IN.
```

- In una clausola WHERE con l'outer join di più di due tabelle, l'operatore (+) può essere applicato solo una volta a una determinata tabella. Nell'esempio seguente, alla tabella SALES non si può fare riferimento con l'operatore (+) in due join successivi.

```
select count(*) from sales, listing, event
where sales.listid(+)=listing.listid and sales.dateid(+)=date.dateid;
ERROR:  A table may be outer joined to at most one other table.
```

- Se la condizione dell'outer join della clausola WHERE confronta una colonna della TABLE2 con una costante, applica l'operatore (+) alla colonna. Se non includi l'operatore, vengono eliminate le righe con l'outer join della TABLE1 che contengono valori null per la colonna con restrizioni. Vedi di seguito la sezione Esempi.

Esempi

La seguente query di join specifica un outer join sinistro delle tabelle SALES e LISTING sulle rispettive colonne LISTID:

```
select count(*)
from sales, listing
where sales.listid = listing.listid(+);
```

```
count
--------
172456
(1 row)
```

La seguente query equivalente produce lo stesso risultato ma utilizza la sintassi di join della clausola FROM:

```
select count(*)
from sales left outer join listing on sales.listid = listing.listid;

count
--------
172456
(1 row)
```

La tabella SALES non contiene i record di tutti gli elenchi della tabella LISTING perché non tutti gli elenchi restituiscono vendite. La seguente query con gli outer join di SALES e LISTING restituisce le righe da LISTING anche quando la tabella SALES non restituisce le vendite per un determinato ID elenco. Le colonne PRICE e COMM, derivate dalla tabella SALES, contengono valori null nel set di risultati per le righe non corrispondenti.

```
select listing.listid, sum(pricepaid) as price,
sum(commission) as comm
from listing, sales
where sales.listid(+) = listing.listid and listing.listid between 1 and 5
group by 1 order by 1;

listid | price  |  comm
--------+--------+--------
1 | 728.00 | 109.20
2 |        |
3 |        |
4 |  76.00 |  11.40
5 | 525.00 |  78.75
(5 rows)
```

Tieni presente che quando viene utilizzato l'operatore join della clausola WHERE, non ha importanza l'ordine delle tabelle nella clausola FROM.

Un esempio di una condizione outer join più complessa nella clausola WHERE è il caso in cui la condizione consiste in un confronto tra due colonne di tabella e un confronto con una costante:

```
where category.catid=event.catid(+) and eventid(+)=796;
```

L'operatore (+) viene utilizzato in due punti: prima nel confronto di uguaglianza tra le tabelle e poi nella condizione di confronto per la colonna EVENTID. Il risultato di questa sintassi è la conservazione delle righe di outer join quando viene valutata la limitazione su EVENTID. Se rimuovi l'operatore (+) dalla restrizione EVENTID, la query considera questa limitazione come un filtro e non come parte della condizione di outer join. A loro volta, le righe dell'outer join che contengono valori null per EVENTID vengono eliminate dal set di risultati.

Ecco una query completa che illustra questo comportamento:

```
select catname, catgroup, eventid
from category, event
where category.catid=event.catid(+) and eventid(+)=796;

catname | catgroup | eventid
-----------+----------+---------
Classical | Concerts |
Jazz | Concerts |
MLB | Sports   |
MLS | Sports   |
Musicals | Shows    | 796
NBA | Sports   |
NFL | Sports   |
NHL | Sports   |
Opera | Shows    |
Plays | Shows    |
Pop | Concerts |
(11 rows)
```

La query equivalente con la sintassi della clausola FROM è la seguente:

```
select catname, catgroup, eventid
from category left join event
on category.catid=event.catid and eventid=796;
```

Se rimuovi il secondo operatore (+) dalla versione della clausola WHERE di questa query, viene restituita solo una riga (la riga dove `eventid=796`).

```
select catname, catgroup, eventid
from category, event
where category.catid=event.catid(+) and eventid=796;

catname | catgroup | eventid
----------+----------+---------
Musicals | Shows    | 796
(1 row)
```

Clausola GROUP BY

La clausola GROUP BY identifica le colonne di raggruppamento per la query. Le colonne di raggruppamento devono essere dichiarate quando la query calcola gli aggregati con le funzioni standard, ad esempio SUM, AVG e COUNT. Per ulteriori informazioni, consulta Funzioni di aggregazione.

Sintassi

```
GROUP BY group_by_clause [, ...]

group_by_clause := {
    expr |
    GROUPING SETS ( () | group_by_clause [, ...] ) |
    ROLLUP ( expr [, ...] ) |
    CUBE ( expr [, ...] )
    }
```

Parametri

expr

L'elenco di colonne o espressioni deve corrispondere all'elenco di espressioni non aggregate dell'elenco di selezione della query. A titolo illustrativo, prendi in considerazione la query semplice riportata di seguito.

```
select listid, eventid, sum(pricepaid) as revenue,
count(qtysold) as numtix
from sales
group by listid, eventid
order by 3, 4, 2, 1
limit 5;
```

```
listid | eventid | revenue | numtix
-------+---------+---------+--------
89397  |      47 |   20.00 |      1
106590 |      76 |   20.00 |      1
124683 |     393 |   20.00 |      1
103037 |     403 |   20.00 |      1
147685 |     429 |   20.00 |      1
(5 rows)
```

In questa query, l'elenco di selezione è composto da due espressioni di aggregazione. La prima utilizza la funzione SUM e la seconda utilizza la funzione COUNT. Le restanti due colonne, LISTID ed EVENTID, devono essere dichiarate come colonne di raggruppamento.

Le espressioni nella clausola GROUP BY possono anche fare riferimento all'elenco di selezione usando numeri ordinali. A titolo illustrativo, l'esempio precedente potrebbe essere abbreviato come segue.

```
select listid, eventid, sum(pricepaid) as revenue,
count(qtysold) as numtix
from sales
group by 1,2
order by 3, 4, 2, 1
limit 5;

listid | eventid | revenue | numtix
-------+---------+---------+--------
89397  |      47 |   20.00 |      1
106590 |      76 |   20.00 |      1
124683 |     393 |   20.00 |      1
103037 |     403 |   20.00 |      1
147685 |     429 |   20.00 |      1
(5 rows)
```

GROUPING SETS/ROLLUP/CUBE

È possibile utilizzare le estensioni di aggregazione GROUPING SETS, ROLLUP e CUBE per eseguire il lavoro di più operazioni GROUP BY in una singola istruzione. Per ulteriori informazioni sulle estensioni di aggregazione e sulle funzioni correlate, consulta Estensioni di aggregazione.

Estensioni di aggregazione

Amazon Redshift supporta estensioni di aggregazione per eseguire il lavoro di più operazioni GROUP BY in un'unica istruzione.

Gli esempi di estensioni di aggregazione utilizzano la tabella `orders`, che contiene i dati di vendita di un'azienda di elettronica. Per creare gli `orders`, procedi come segue.

```
CREATE TABLE ORDERS (
    ID INT,
    PRODUCT CHAR(20),
    CATEGORY CHAR(20),
    PRE_OWNED CHAR(1),
    COST DECIMAL
);

INSERT INTO ORDERS VALUES
    (0, 'laptop',       'computers',    'T', 1000),
    (1, 'smartphone',   'cellphones',   'T', 800),
    (2, 'smartphone',   'cellphones',   'T', 810),
    (3, 'laptop',       'computers',    'F', 1050),
    (4, 'mouse',        'computers',    'F', 50);
```

GROUPING SETS

Calcola uno o più set di raggruppamento in una singola istruzione. Un set di raggruppamento è l'insieme di una singola clausola GROUP BY, un set di 0 o più colonne in base al quale è possibile raggruppare il set di risultati di una query. GROUP BY GROUPING SETS equivale all'esecuzione di una query UNION ALL su un set di risultati raggruppato in colonne diverse. Ad esempio, GROUP BY GROUP SETS((a), (b)) è equivalente a GROUP BY a UNION ALL GROUP BY b.

L'esempio seguente restituisce il costo dei prodotti nella tabella degli ordini raggruppati in base alle categorie dei prodotti e al tipo di prodotti venduti.

```
SELECT category, product, sum(cost) as total
FROM orders
GROUP BY GROUPING SETS(category, product);

       category       |       product        | total
----------------------+----------------------+-------
```

```
 computers            |                      |  2100
 cellphones           |                      |  1610
                      | laptop               |  2050
                      | smartphone           |  1610
                      | mouse                |    50

(5 rows)
```

ROLLUP

Presuppone una gerarchia in cui le colonne precedenti sono considerate le colonne padri delle colonne successive. ROLLUP raggruppa i dati in base alle colonne fornite, restituendo righe di subtotali aggiuntive che rappresentano i totali in tutti i livelli di colonne di raggruppamento, oltre alle righe raggruppate. Ad esempio, puoi utilizzare GROUP BY ROLLUP((a), (b)) per restituire un set di risultati raggruppato prima per a, poi per b supponendo che b sia una sottosezione di a. ROLLUP restituisce anche una riga con l'intero set di risultati senza colonne di raggruppamento.

GROUP BY ROLLUP((a), (b)) è equivalente a GROUP BY GROUPING SETS((a,b), (a), ()).

L'esempio seguente restituisce il costo dei prodotti nella tabella degli ordini raggruppati prima per categoria e poi per prodotto, con product come sezione della categoria.

```
SELECT category, product, sum(cost) as total
FROM orders
GROUP BY ROLLUP(category, product) ORDER BY 1,2;

       category       |       product        | total
----------------------+----------------------+-------
 cellphones           | smartphone           |  1610
 cellphones           |                      |  1610
 computers            | laptop               |  2050
 computers            | mouse                |    50
 computers            |                      |  2100
                      |                      |  3710
(6 rows)
```

CUBE

Raggruppa i dati in base alle colonne fornite, restituendo righe di subtotali aggiuntive che rappresentano i totali in tutti i livelli di colonne di raggruppamento, oltre alle righe raggruppate. CUBE restituisce le stesse righe di ROLLUP, ma aggiunge ulteriori righe di subtotali per ogni combinazione

di colonne di raggruppamento non previste da ROLLUP. Ad esempio, è possibile utilizzare GROUP BY CUBE((a), (b)) per restituire un set di risultati raggruppato prima per a, poi per b, supponendo che b sia una sottosezione di a, quindi solo per b. CUBE restituisce anche una riga con l'intero set di risultati senza colonne di raggruppamento.

GROUP BY CUBE((a), (b)) è equivalente a GROUP BY GROUPING SETS((a, b), (a), (b), ()).

L'esempio seguente restituisce il costo dei prodotti nella tabella degli ordini raggruppati prima per categoria e poi per prodotto, con product come sezione della categoria. A differenza dell'esempio precedente per ROLLUP, l'istruzione restituisce risultati per ogni combinazione di colonne di raggruppamento.

```
SELECT category, product, sum(cost) as total
FROM orders
GROUP BY CUBE(category, product) ORDER BY 1,2;

       category       |       product        | total
----------------------+----------------------+-------
 cellphones           | smartphone           |  1610
 cellphones           |                      |  1610
 computers            | laptop               |  2050
 computers            | mouse                |    50
 computers            |                      |  2100
                      | laptop               |  2050
                      | mouse                |    50
                      | smartphone           |  1610
                      |                      |  3710
(9 rows)
```

Funzioni GROUPING/GROUPING_ID

ROLLUP e CUBE aggiungono valori NULL al set di risultati per indicare le righe di subtotali. Ad esempio, GROUP BY ROLLUP((a), (b)) restituisce una o più righe con un valore NULL nella colonna di raggruppamento b per indicare che si tratta di subtotali dei campi nella colonna di raggruppamento a. Questi valori NULL servono solo a soddisfare il formato delle tuple restituite.

Quando si eseguono operazioni GROUP BY con ROLLUP e CUBE su relazioni che memorizzano i valori NULL, è possibile produrre set di risultati con righe che sembrano avere colonne di raggruppamento identiche. Tornando all'esempio precedente, se la colonna di raggruppamento b contiene un valore NULL memorizzato, GROUP BY ROLLUP((a), (b)) restituisce una riga con un valore NULL nella colonna di raggruppamento b che non è un subtotale.

Per distinguere tra i valori NULL creati da ROLLUP e CUBE e i valori NULL memorizzati nelle tabelle stesse, è possibile utilizzare la funzione GROUPING o il relativo alias GROUPING_ID. GROUPING accetta un singolo set di raggruppamento come argomento e per ogni riga del set di risultati restituisce un valore di 0 o 1 bit corrispondente alla colonna di raggruppamento in quella posizione, quindi converte tale valore in un numero intero. Se il valore in quella posizione è un valore NULL creato da un'estensione di aggregazione, GROUPING restituisce 1. Restituisce 0 per tutti gli altri valori, inclusi i valori NULL memorizzati.

Ad esempio, GROUPING(category, product) può restituire i seguenti valori per una determinata riga, a seconda dei valori della colonna di raggruppamento per quella riga. Ai fini di questo esempio, tutti i valori NULL nella tabella sono creati da un'estensione di aggregazione.

Colonna della categoria	Colonna del prodotto	Valore in bit della funzione GROUPING	Valore decimale
Non NULL	Non NULL	00	0
Non NULL	NULL	01	1
NULL	Non NULL	10	2
NULL	NULL	11	3

Le funzioni di GROUPING vengono mostrate nella parte dell'elenco SELECT della query nel seguente formato.

```
SELECT ... [GROUPING( expr )...] ...
  GROUP BY ... {CUBE | ROLLUP| GROUPING SETS} ( expr ) ...
```

L'esempio seguente è analogo all'esempio precedente per CUBE, ma con l'aggiunta di funzioni GROUPING per i relativi set di raggruppamento.

```
SELECT category, product,
       GROUPING(category) as grouping0,
       GROUPING(product) as grouping1,
       GROUPING(category, product) as grouping2,
       sum(cost) as total
```

```
FROM orders
GROUP BY CUBE(category, product) ORDER BY 3,1,2;

       category       |       product        | grouping0 | grouping1 | grouping2 |
 total
----------------------+----------------------+-----------+-----------+-----------
+-------
 cellphones           | smartphone           |         0 |         0 |         0 |
 1610
 cellphones           |                      |         0 |         1 |         1 |
 1610
 computers            | laptop               |         0 |         0 |         0 |
 2050
 computers            | mouse                |         0 |         0 |         0 |
 50
 computers            |                      |         0 |         1 |         1 |
 2100
                      | laptop               |         1 |         0 |         2 |
 2050
                      | mouse                |         1 |         0 |         2 |
 50
                      | smartphone           |         1 |         0 |         2 |
 1610
                      |                      |         1 |         1 |         3 |
 3710
(9 rows)
```

ROLLUP e CUBE parziali

È possibile eseguire operazioni ROLLUP e CUBE con solo una parte dei subtotali.

La sintassi per le operazioni parziali di ROLLUP e CUBE è la seguente.

```
GROUP BY expr1, { ROLLUP | CUBE }(expr2, [, ...])
```

Qui, la clausola GROUP BY crea solo righe di subtotali a livello di expr2 e successivi.

Gli esempi seguenti mostrano le operazioni parziali di ROLLUP e CUBE nella tabella degli ordini, raggruppate in base allo stato di usato o meno di un prodotto; le operazioni di ROLLUP e CUBE vengono quindi eseguite nelle colonne della categoria e del prodotto.

```
SELECT pre_owned, category, product,
```

```
       GROUPING(category, product, pre_owned) as group_id,
       sum(cost) as total
FROM orders
GROUP BY pre_owned, ROLLUP(category, product) ORDER BY 4,1,2,3;

 pre_owned |       category       |       product        | group_id | total
-----------+----------------------+----------------------+----------+-------
 F         | computers            | laptop               |        0 |  1050
 F         | computers            | mouse                |        0 |    50
 T         | cellphones           | smartphone           |        0 |  1610
 T         | computers            | laptop               |        0 |  1000
 F         | computers            |                      |        2 |  1100
 T         | cellphones           |                      |        2 |  1610
 T         | computers            |                      |        2 |  1000
 F         |                      |                      |        6 |  1100
 T         |                      |                      |        6 |  2610
(9 rows)

SELECT pre_owned, category, product,
       GROUPING(category, product, pre_owned) as group_id,
       sum(cost) as total
FROM orders
GROUP BY pre_owned, CUBE(category, product) ORDER BY 4,1,2,3;

 pre_owned |       category       |       product        | group_id | total
-----------+----------------------+----------------------+----------+-------
 F         | computers            | laptop               |        0 |  1050
 F         | computers            | mouse                |        0 |    50
 T         | cellphones           | smartphone           |        0 |  1610
 T         | computers            | laptop               |        0 |  1000
 F         | computers            |                      |        2 |  1100
 T         | cellphones           |                      |        2 |  1610
 T         | computers            |                      |        2 |  1000
 F         |                      | laptop               |        4 |  1050
 F         |                      | mouse                |        4 |    50
 T         |                      | laptop               |        4 |  1000
 T         |                      | smartphone           |        4 |  1610
 F         |                      |                      |        6 |  1100
 T         |                      |                      |        6 |  2610
(13 rows)
```

Poiché la colonna dei prodotti usati non è inclusa nelle operazioni ROLLUP e CUBE, non esiste una riga totale complessiva che includa tutte le altre righe.

Raggruppamento concatenato

È possibile concatenare più clausole GROUPING SETS/ROLLUP/CUBE per calcolare diversi livelli di subtotali. I raggruppamenti concatenati restituiscono il prodotto cartesiano dei set di raggruppamento forniti.

La sintassi per concatenare le clausole GROUPING SETS/ROLLUP/CUBE è la seguente.

```
GROUP BY {ROLLUP|CUBE|GROUPING SETS}(expr1[, ...]),
         {ROLLUP|CUBE|GROUPING SETS}(expr1[, ...])[, ...]
```

L'esempio seguente mostra come un piccolo raggruppamento concatenato possa produrre un notevole set di risultati finali.

```
SELECT pre_owned, category, product,
       GROUPING(category, product, pre_owned) as group_id,
       sum(cost) as total
FROM orders
GROUP BY CUBE(category, product), GROUPING SETS(pre_owned, ())
ORDER BY 4,1,2,3;

 pre_owned |       category       |       product        | group_id | total
-----------+----------------------+----------------------+----------+-------
 F         | computers            | laptop               |        0 |  1050
 F         | computers            | mouse                |        0 |    50
 T         | cellphones           | smartphone           |        0 |  1610
 T         | computers            | laptop               |        0 |  1000
           | cellphones           | smartphone           |        1 |  1610
           | computers            | laptop               |        1 |  2050
           | computers            | mouse                |        1 |    50
 F         | computers            |                      |        2 |  1100
 T         | cellphones           |                      |        2 |  1610
 T         | computers            |                      |        2 |  1000
           | cellphones           |                      |        3 |  1610
           | computers            |                      |        3 |  2100
 F         |                      | laptop               |        4 |  1050
 F         |                      | mouse                |        4 |    50
 T         |                      | laptop               |        4 |  1000
 T         |                      | smartphone           |        4 |  1610
           |                      | laptop               |        5 |  2050
           |                      | mouse                |        5 |    50
           |                      | smartphone           |        5 |  1610
 F         |                      |                      |        6 |  1100
```

```
 T         |                     |                     |        6 |  2610
           |                     |                     |        7 |  3710
(22 rows)
```

Raggruppamento nidificato

È possibile utilizzare le operazioni GROUPING SETS/ROLLUP/CUBE come GROUPING SETS espr per formare un raggruppamento nidificato. Il sottoraggruppamento all'interno dei GROUPING SETS nidificati viene appiattito.

La sintassi per il raggruppamento nidificato è la seguente:

```
GROUP BY GROUPING SETS({ROLLUP|CUBE|GROUPING SETS}(expr[, ...])[, ...])
```

Analizza l'esempio seguente.

```
SELECT category, product, pre_owned,
       GROUPING(category, product, pre_owned) as group_id,
       sum(cost) as total
FROM orders
GROUP BY GROUPING SETS(ROLLUP(category), CUBE(product, pre_owned))
ORDER BY 4,1,2,3;

       category       |       product        | pre_owned | group_id | total
----------------------+----------------------+-----------+----------+-------
 cellphones           |                      |           |        3 |  1610
 computers            |                      |           |        3 |  2100
                      | laptop               | F         |        4 |  1050
                      | laptop               | T         |        4 |  1000
                      | mouse                | F         |        4 |    50
                      | smartphone           | T         |        4 |  1610
                      | laptop               |           |        5 |  2050
                      | mouse                |           |        5 |    50
                      | smartphone           |           |        5 |  1610
                      |                      | F         |        6 |  1100
                      |                      | T         |        6 |  2610
                      |                      |           |        7 |  3710
                      |                      |           |        7 |  3710
(13 rows)
```

È importante notare che, poiché sia ROLLUP(category) che CUBE(product, pre_owned) contengono il set di raggruppamento (), la riga che rappresenta il totale complessivo viene duplicata.

Note per l'utilizzo

- La clausola GROUP BY supporta fino a 64 set di raggruppamento. Nel caso di ROLLUP e CUBE o di una combinazione di GROUPING SETS, ROLLUP e CUBE, questa limitazione si applica al numero implicito di set di raggruppamento. Ad esempio, GROUP BY CUBE((a), (b)) conta per 4 set di raggruppamento, non 2.
- Non è possibile utilizzare le costanti come colonne di raggruppamento quando si utilizzano le estensioni di aggregazione.
- Non è possibile creare un set di raggruppamento che contiene colonne duplicate.

Clausola HAVING

La clausola HAVING applica una condizione al set di risultati raggruppati intermedi restituiti da una query.

Sintassi

```
[ HAVING condition ]
```

Ad esempio, puoi limitare i risultati di una funzione SUM:

```
having sum(pricepaid) >10000
```

La condizione HAVING viene applicata dopo che tutte le condizioni della clausola WHERE sono state applicate e le operazioni GROUP BY sono state completate.

La condizione stessa assume lo stesso formato di qualsiasi condizione della clausola WHERE.

Note per l'utilizzo

- Qualsiasi colonna a cui viene fatto riferimento in una condizione della clausola HAVING deve essere una colonna di raggruppamento o una colonna che fa riferimento al risultato di una funzione di aggregazione.
- In una clausola HAVING, non è possibile specificare:
  - Un numero ordinale che fa riferimento a una voce di elenco selezionata. Solo le clausole GROUP BY e ORDER BY accettano numeri ordinali.

Esempi

La seguente query calcola le vendite totali dei biglietti per tutti gli eventi in base al nome, quindi elimina gli eventi in cui le vendite totali erano inferiori a $800.000. La condizione HAVING viene applicata ai risultati della funzione di aggregazione nell'elenco di selezione: `sum(pricepaid)`.

```
select eventname, sum(pricepaid)
from sales join event on sales.eventid = event.eventid
group by 1
having sum(pricepaid) > 800000
order by 2 desc, 1;

eventname        |    sum
-----------------+-----------
Mamma Mia!       | 1135454.00
Spring Awakening |  972855.00
The Country Girl |  910563.00
Macbeth          |  862580.00
Jersey Boys      |  811877.00
Legally Blonde   |  804583.00
```

La seguente query calcola un set di risultati simile. In questo caso, tuttavia, la condizione HAVING viene applicata a un'aggregazione che non è specificata nell'elenco di selezione: `sum(qtysold)`. Gli eventi che non hanno venduto più di 2.000 biglietti sono stati eliminati dal risultato finale.

```
select eventname, sum(pricepaid)
from sales join event on sales.eventid = event.eventid
group by 1
having sum(qtysold) >2000
order by 2 desc, 1;

eventname        |    sum
-----------------+-----------
Mamma Mia!       | 1135454.00
Spring Awakening |  972855.00
The Country Girl |  910563.00
Macbeth          |  862580.00
Jersey Boys      |  811877.00
Legally Blonde   |  804583.00
Chicago          |  790993.00
Spamalot         |  714307.00
```

La seguente query calcola le vendite totali dei biglietti per tutti gli eventi in base al nome, quindi elimina gli eventi in cui le vendite totali erano inferiori a $800.000. La condizione HAVING viene applicata ai risultati della funzione di aggregazione nell'elenco di selezione utilizzando l'alias pp for. `sum(pricepaid)`

```
select eventname, sum(pricepaid) as pp
from sales join event on sales.eventid = event.eventid
group by 1
having pp > 800000
order by 2 desc, 1;

eventname        |    pp
-----------------+-----------
Mamma Mia!       | 1135454.00
Spring Awakening |  972855.00
The Country Girl |  910563.00
Macbeth          |  862580.00
Jersey Boys      |  811877.00
Legally Blonde   |  804583.00
```

Clausola QUALIFY

La clausola QUALIFY filtra i risultati di una funzione finestra calcolata in precedenza in base alle condizioni di ricerca specificate dall'utente. È possibile utilizzare la clausola per applicare condizioni di filtro al risultato di una funzione finestra senza utilizzare una sottoquery.

È simile alla clausola HAVING, che applica una condizione per filtrare ulteriormente le righe da una clausola WHERE. La differenza tra QUALIFY e HAVING è che i risultati filtrati dalla clausola QUALIFY potrebbero essere basati sul risultato dell'esecuzione delle funzioni finestra sui dati. Puoi utilizzare entrambe le clausole QUALIFY e HAVING in un'unica query.

Sintassi

```
QUALIFY condition
```

> **Note**
>
> Se utilizzi la clausola QUALIFY direttamente dopo la clausola FROM, il nome della relazione FROM deve avere un alias specificato prima della clausola QUALIFY.

Esempi

Gli esempi in questa sezione utilizzano i dati di esempio riportati di seguito.

```
create table store_sales (ss_sold_date date, ss_sold_time time,
               ss_item text, ss_sales_price float);
insert into store_sales values ('2022-01-01', '09:00:00', 'Product 1', 100.0),
                               ('2022-01-01', '11:00:00', 'Product 2', 500.0),
                               ('2022-01-01', '15:00:00', 'Product 3', 20.0),
                               ('2022-01-01', '17:00:00', 'Product 4', 1000.0),
                               ('2022-01-01', '18:00:00', 'Product 5', 30.0),
                               ('2022-01-02', '10:00:00', 'Product 6', 5000.0),
                               ('2022-01-02', '16:00:00', 'Product 7', 5.0);
```

L'esempio seguente mostra come trovare i due articoli più costosi venduti dopo le 12:00 di ogni giorno.

```
SELECT *
FROM store_sales ss
WHERE ss_sold_time > time '12:00:00'
QUALIFY row_number()
OVER (PARTITION BY ss_sold_date ORDER BY ss_sales_price DESC) <= 2


 ss_sold_date | ss_sold_time |  ss_item  | ss_sales_price
--------------+--------------+-----------+----------------
 2022-01-01   | 17:00:00     | Product 4 |           1000
 2022-01-01   | 18:00:00     | Product 5 |             30
 2022-01-02   | 16:00:00     | Product 7 |              5
```

Potrai quindi trovare l'ultimo articolo venduto ogni giorno.

```
SELECT *
FROM store_sales ss
QUALIFY last_value(ss_item)
OVER (PARTITION BY ss_sold_date ORDER BY ss_sold_time ASC
      ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) = ss_item;

ss_sold_date | ss_sold_time |  ss_item  | ss_sales_price
--------------+--------------+-----------+----------------
 2022-01-01   | 18:00:00     | Product 5 |             30
 2022-01-02   | 16:00:00     | Product 7 |              5
```

L'esempio seguente restituisce gli stessi record della query precedente, l'ultimo articolo venduto ogni giorno, ma non utilizza la clausola QUALIFY.

```
SELECT * FROM (
  SELECT *,
  last_value(ss_item)
  OVER (PARTITION BY ss_sold_date ORDER BY ss_sold_time ASC
        ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) ss_last_item
  FROM store_sales ss
)
WHERE ss_last_item = ss_item;

 ss_sold_date | ss_sold_time |  ss_item  | ss_sales_price | ss_last_item
--------------+--------------+-----------+----------------+--------------
 2022-01-02   | 16:00:00     | Product 7 |              5 | Product 7
 2022-01-01   | 18:00:00     | Product 5 |             30 | Product 5
```

UNION, INTERSECT ed EXCEPT

Argomenti

- Sintassi
- Parametri
- Ordine di valutazione degli operatori di definizione
- Note per l'utilizzo
- Query UNION di esempio
- Query UNION ALL di esempio
- Query INTERSECT di esempio
- Query EXCEPT di esempio

Gli operatori di definizione UNION, INTERSECT ed EXCEPT vengono utilizzati per confrontare e unire i risultati di due espressioni di query separate. Ad esempio, se vuoi sapere quali utenti di un sito web sono sia acquirenti che venditori ma i loro nomi utente sono memorizzati in colonne o tabelle separate, puoi trovare l'intersezione di questi due tipi di utenti. Se vuoi sapere quali utenti del sito sono acquirenti ma non venditori, puoi utilizzare l'operatore EXCEPT per trovare la difference tra i due elenchi di utenti. Se vuoi creare l'elenco di tutti gli utenti, indipendentemente dal ruolo, puoi utilizzare l'operatore UNION.

Sintassi

```
query
{ UNION [ ALL ] | INTERSECT | EXCEPT | MINUS }
query
```

Parametri

query

Espressione di query che corrisponde, sotto forma di elenco di selezione, a una seconda espressione di query che segue l'operatore UNION, INTERSECT o EXCEPT. Le due espressioni devono contenere lo stesso numero di colonne di output con tipi di dati compatibili; in caso contrario, i due set di risultati non possono essere confrontati e uniti. Le operazioni di definizione non consentono la conversione implicita tra diverse categorie di tipi di dati. Per ulteriori informazioni, consultare Conversione e compatibilità dei tipi.

Puoi creare query contenenti un numero illimitato di espressioni di query e collegarle agli operatori UNION, INTERSECT ed EXCEPT in qualsiasi combinazione. Ad esempio, la seguente struttura di query è valida, assumendo che le tabelle T1, T2 e T3 contengano set di colonne compatibili:

```
select * from t1
union
select * from t2
except
select * from t3
order by c1;
```

UNION

Operazione di definizione che restituisce le righe da due espressioni di query, indipendentemente dal fatto che le righe derivino da una o entrambe le espressioni.

INTERSECT

Operazione di definizione che restituisce le righe che derivano da due espressioni di query. Le righe che non vengono restituite da entrambe le espressioni vengono scartate.

EXCEPT | MINUS

Operazione di definizione che restituisce le righe che derivano da una delle due espressioni di query. Per qualificarsi per il risultato, le righe devono esistere nella prima tabella dei risultati ma non nella seconda. MINUS ed EXCEPT sono sinonimi esatti.

ALL

La parola chiave ALL conserva tutte le righe duplicate prodotte da UNION. Il comportamento predefinito quando la parola chiave ALL non viene utilizzata è di scartare questi duplicati. INTERSECT ALL, EXCEPT ALL e MINUS ALL non sono supportati.

Ordine di valutazione degli operatori di definizione

Gli operatori di definizione UNION ed EXCEPT sono associativi a sinistra. Se non si specificano le parentesi per influenzare l'ordine di precedenza, una combinazione di questi operatori di definizione viene valutata da sinistra a destra. Ad esempio, nella seguente query, l'operazione UNION di T1 e T2 viene valutata per prima, quindi l'operazione EXCEPT viene eseguita sul risultato di UNION:

```
select * from t1
union
select * from t2
except
select * from t3
order by c1;
```

L'operatore INTERSECT ha la precedenza sugli operatori UNION ed EXCEPT quando una combinazione di operatori viene utilizzata nella stessa query. Ad esempio, la seguente query valuta l'intersezione di T2 e T3, quindi l'unione del risultato con T1:

```
select * from t1
union
select * from t2
intersect
select * from t3
order by c1;
```

Aggiungendo le parentesi, puoi applicare un diverso ordine di valutazione. Nel seguente caso, il risultato dell'unione di T1 e T2 viene intersecato con T3 e la query produce probabilmente un risultato diverso.

```
(select * from t1
union
select * from t2)
intersect
(select * from t3)
order by c1;
```

Note per l'utilizzo

- I nomi di colonna restituiti nel risultato di una query dell'operazione di definizione sono i nomi di colonna (o alias) delle tabelle nella prima espressione di query. Poiché questi nomi di colonne sono potenzialmente fuorvianti, in quanto i valori della colonna derivano da tabelle su entrambi i lati dell'operatore di definizione, puoi fornire alias significativi per il set di risultati.
- Un'espressione di query che precede un operatore di definizione non deve contenere una clausola ORDER BY. Una clausola ORDER BY produce risultati ordinati significativi solo quando viene utilizzata alla fine di una query che contiene operatori di definizione. In questo caso, la clausola ORDER BY si applica ai risultati finali di tutte le operazioni di definizione. La query più esterna può contenere anche clausole LIMIT e OFFSET standard.
- Quando le query dell'operatore di definizione restituiscono risultati decimali, le colonne dei risultati corrispondenti vengono elevate per restituire la stessa precisione e scala. Ad esempio, nella seguente query, dove T1.REVENUE è una colonna DECIMAL (10,2) e T2.REVENUE è una colonna DECIMAL (8,4), il risultato decimale viene elevato a DECIMAL (12,4):

  ```
  select t1.revenue union select t2.revenue;
  ```

  La scala è 4 perché è la scala massima delle due colonne. La precisione è 12 perché T1.REVENUE richiede 8 cifre a sinistra del punto decimale (12 - 4 = 8). Questo tipo di elevazione garantisce che tutti i valori di entrambi i lati dell'UNION si adattino al risultato. Per i valori a 64 bit, la precisione massima del risultato è 19 e la scala del risultato massimo è 18. Per i valori a 128 bit, la precisione massima del risultato è 38 e la scala del risultato massimo è 37.

  Se il tipo di dati risultante supera limiti di precisione e scala di Amazon Redshift, la query restituisce un errore.
- Per le operazioni di definizione, due righe vengono considerate identiche se, per ciascuna coppia di colonne corrispondente, i due valori di dati sono uguali o entrambi NULL. Ad esempio, se le tabelle T1 e T2 contengono entrambe una colonna e una riga e tale riga è NULL in entrambe le tabelle, un'operazione INTERSECT su quelle tabelle restituisce tale riga.

Query UNION di esempio

Nella seguente query UNION, le righe nella tabella SALES vengono unite alle righe nella tabella LISTING. Tre colonne compatibili sono selezionate da ciascuna tabella; in questo caso, le colonne corrispondenti hanno gli stessi nomi e tipi di dati.

Il set di risultati finale è ordinato dalla prima colonna nella tabella LISTING e limitato alle 5 righe con il valore LISTID più alto.

```
select listid, sellerid, eventid from listing
union select listid, sellerid, eventid from sales
order by listid, sellerid, eventid desc limit 5;

listid | sellerid | eventid
--------+----------+---------
1 |    36861 |    7872
2 |    16002 |    4806
3 |    21461 |    4256
4 |     8117 |    4337
5 |     1616 |    8647
(5 rows)
```

L'esempio seguente mostra come è possibile aggiungere un valore letterale all'output di una query UNION in modo da poter vedere quale espressione di query ha prodotto ogni riga nel set di risultati. La query identifica le righe dalla prima espressione di query come "B" (per gli acquirenti) e le righe dalla seconda espressione di query come "S" (per i venditori).

La query identifica acquirenti e venditori per transazioni di biglietti che costano almeno $10.000. L'unica differenza tra le due espressioni di query su entrambi i lati dell'operatore UNION è la colonna di collegamento per la tabella SALES.

```
select listid, lastname, firstname, username,
pricepaid as price, 'S' as buyorsell
from sales, users
where sales.sellerid=users.userid
and pricepaid >=10000
union
select listid, lastname, firstname, username, pricepaid,
'B' as buyorsell
from sales, users
where sales.buyerid=users.userid
```

```
and pricepaid >=10000
order by 1, 2, 3, 4, 5;

listid | lastname | firstname | username |   price   | buyorsell
--------+----------+-----------+----------+-----------+-----------
209658 | Lamb     | Colette   | VOR15LYI |  10000.00 | B
209658 | West     | Kato      | ELU81XAA |  10000.00 | S
212395 | Greer    | Harlan    | GXO71KOC |  12624.00 | S
212395 | Perry    | Cora      | YWR73YNZ |  12624.00 | B
215156 | Banks    | Patrick   | ZNQ69CLT |  10000.00 | S
215156 | Hayden   | Malachi   | BBG56AKU |  10000.00 | B
(6 rows)
```

L'esempio seguente utilizza un operatore UNION ALL perché le righe duplicate, se trovate, devono essere conservate nel risultato. Per una serie specifica di ID evento, la query restituisce 0 o più righe per ogni vendita associata a ciascun evento e 0 o 1 riga per ciascun elenco di quell'evento. Gli ID evento sono unici per ogni riga nelle tabelle LISTING ed EVENT, ma potrebbero esserci più vendite per la stessa combinazione di ID evento ed elenco nella tabella SALES.

La terza colonna nel set di risultati identifica l'origine della riga. Se proviene dalla tabella SALES, è contrassegnata con "Yes" nella colonna SALESROW. SALESROW è un alias per SALES.LISTID. Se la riga proviene dalla tabella LISTING, è contrassegnata con "No" nella colonna SALESROW.

In questo caso, il set di risultati è costituito da tre righe di vendita per l'elenco 500, evento 7787. In altre parole, sono state eseguite tre diverse transazioni per l'elenco e la combinazione di eventi. Gli altri due elenchi, 501 e 502, non hanno prodotto alcuna vendita, quindi l'unica riga prodotta dalla query per questi ID elenco proviene dalla tabella LISTING (SALESROW = 'No').

```
select eventid, listid, 'Yes' as salesrow
from sales
where listid in(500,501,502)
union all
select eventid, listid, 'No'
from listing
where listid in(500,501,502)
order by listid asc;

eventid | listid | salesrow
---------+--------+----------
7787 |    500 | No
7787 |    500 | Yes
7787 |    500 | Yes
```

```
7787 |    500 | Yes
6473 |    501 | No
5108 |    502 | No
(6 rows)
```

Se esegui la stessa query senza la parola chiave ALL, il risultato conserva solo una delle transazioni di vendita.

```
select eventid, listid, 'Yes' as salesrow
from sales
where listid in(500,501,502)
union
select eventid, listid, 'No'
from listing
where listid in(500,501,502)
order by listid asc;

eventid | listid | salesrow
---------+--------+----------
7787 |    500 | No
7787 |    500 | Yes
6473 |    501 | No
5108 |    502 | No
(4 rows)
```

Query UNION ALL di esempio

L'esempio seguente utilizza un operatore UNION ALL perché le righe duplicate, se trovate, devono essere conservate nel risultato. Per una serie specifica di ID evento, la query restituisce 0 o più righe per ogni vendita associata a ciascun evento e 0 o 1 riga per ciascun elenco di quell'evento. Gli ID evento sono unici per ogni riga nelle tabelle LISTING ed EVENT, ma potrebbero esserci più vendite per la stessa combinazione di ID evento ed elenco nella tabella SALES.

La terza colonna nel set di risultati identifica l'origine della riga. Se proviene dalla tabella SALES, è contrassegnata con "Yes" nella colonna SALESROW. SALESROW è un alias per SALES.LISTID. Se la riga proviene dalla tabella LISTING, è contrassegnata con "No" nella colonna SALESROW.

In questo caso, il set di risultati è costituito da tre righe di vendita per l'elenco 500, evento 7787. In altre parole, sono state eseguite tre diverse transazioni per l'elenco e la combinazione di eventi. Gli altri due elenchi, 501 e 502, non hanno prodotto alcuna vendita, quindi l'unica riga prodotta dalla query per questi ID elenco proviene dalla tabella LISTING (SALESROW = 'No').

```
select eventid, listid, 'Yes' as salesrow
from sales
where listid in(500,501,502)
union all
select eventid, listid, 'No'
from listing
where listid in(500,501,502)
order by listid asc;

eventid | listid | salesrow
---------+--------+----------
7787 |    500 | No
7787 |    500 | Yes
7787 |    500 | Yes
7787 |    500 | Yes
6473 |    501 | No
5108 |    502 | No
(6 rows)
```

Se esegui la stessa query senza la parola chiave ALL, il risultato conserva solo una delle transazioni di vendita.

```
select eventid, listid, 'Yes' as salesrow
from sales
where listid in(500,501,502)
union
select eventid, listid, 'No'
from listing
where listid in(500,501,502)
order by listid asc;

eventid | listid | salesrow
---------+--------+----------
7787 |    500 | No
7787 |    500 | Yes
6473 |    501 | No
5108 |    502 | No
(4 rows)
```

Query INTERSECT di esempio

Confronta il seguente esempio con il primo esempio di UNION. L'unica differenza tra i due esempi è l'operatore di definizione che viene utilizzato, ma i risultati sono molto diversi. Solo una delle righe è uguale:

```
235494 |    23875 |    8771
```

Questa è l'unica riga del risultato limitato di 5 righe trovata in entrambe le tabelle.

```
select listid, sellerid, eventid from listing
intersect
select listid, sellerid, eventid from sales
order by listid desc, sellerid, eventid
limit 5;

listid | sellerid | eventid
--------+----------+---------
235494 |    23875 |    8771
235482 |     1067 |    2667
235479 |     1589 |    7303
235476 |    15550 |     793
235475 |    22306 |    7848
(5 rows)
```

La seguente query trova gli eventi (per i quali sono stati venduti i biglietti) che si sono verificati nelle sedi di New York City e Los Angeles a marzo. La differenza tra le due espressioni di query è il vincolo sulla colonna VENUECITY.

```
select distinct eventname from event, sales, venue
where event.eventid=sales.eventid and event.venueid=venue.venueid
and date_part(month,starttime)=3 and venuecity='Los Angeles'
intersect
select distinct eventname from event, sales, venue
where event.eventid=sales.eventid and event.venueid=venue.venueid
and date_part(month,starttime)=3 and venuecity='New York City'
order by eventname asc;

eventname
----------------------------
A Streetcar Named Desire
Dirty Dancing
```

```
Electra
Running with Annalise
Hairspray
Mary Poppins
November
Oliver!
Return To Forever
Rhinoceros
South Pacific
The 39 Steps
The Bacchae
The Caucasian Chalk Circle
The Country Girl
Wicked
Woyzeck
(16 rows)
```

Query EXCEPT di esempio

La tabella CATEGORY nel database TICKIT contiene le seguenti 11 righe:

```
 catid | catgroup |  catname  |                  catdesc
-------+----------+-----------+--------------------------------------------
   1   | Sports   | MLB       | Major League Baseball
   2   | Sports   | NHL       | National Hockey League
   3   | Sports   | NFL       | National Football League
   4   | Sports   | NBA       | National Basketball Association
   5   | Sports   | MLS       | Major League Soccer
   6   | Shows    | Musicals  | Musical theatre
   7   | Shows    | Plays     | All non-musical theatre
   8   | Shows    | Opera     | All opera and light opera
   9   | Concerts | Pop       | All rock and pop music concerts
  10   | Concerts | Jazz      | All jazz singers and bands
  11   | Concerts | Classical | All symphony, concerto, and choir concerts
(11 rows)
```

Supponi che una tabella CATEGORY_STAGE (una tabella di gestione temporanea) contenga una riga aggiuntiva:

```
 catid | catgroup |  catname  |                  catdesc
-------+----------+-----------+--------------------------------------------
1 | Sports   | MLB       | Major League Baseball
2 | Sports   | NHL       | National Hockey League
```

```
3 | Sports   | NFL       | National Football League
4 | Sports   | NBA       | National Basketball Association
5 | Sports   | MLS       | Major League Soccer
6 | Shows    | Musicals  | Musical theatre
7 | Shows    | Plays     | All non-musical theatre
8 | Shows    | Opera     | All opera and light opera
9 | Concerts | Pop       | All rock and pop music concerts
10 | Concerts | Jazz      | All jazz singers and bands
11 | Concerts | Classical | All symphony, concerto, and choir concerts
12 | Concerts | Comedy    | All stand up comedy performances
(12 rows)
```

Restituisce la differenza tra le due tabelle. In altre parole, restituisce le righe che si trovano nella tabella CATEGORY_STAGE ma non nella tabella CATEGORY:

```
select * from category_stage
except
select * from category;

catid | catgroup | catname |             catdesc
-------+----------+---------+----------------------------------
12 | Concerts | Comedy  | All stand up comedy performances
(1 row)
```

La seguente query equivalente utilizza il sinonimo MINUS.

```
select * from category_stage
minus
select * from category;

catid | catgroup | catname |             catdesc
-------+----------+---------+----------------------------------
12 | Concerts | Comedy  | All stand up comedy performances
(1 row)
```

Se inverti l'ordine delle espressioni SELECT, la query non restituisce alcuna riga.

Clausola ORDER BY

Argomenti

- Sintassi

La clausola ORDER BY ordina il set di risultati di una query.

Sintassi

```
[ ORDER BY expression [ ASC | DESC ] ]
[ NULLS FIRST | NULLS LAST ]
[ LIMIT { count | ALL } ]
[ OFFSET start ]
```

Parametri

espressione

Espressione che definisce l'ordinamento del set di risultati della query, in genere specificando una o più colonne nell'elenco di selezione. I risultati vengono restituiti in base all'ordinamento binario UTF-8. È anche possibile specificare:

- Colonne che non si trovano nell'elenco di selezione
- Espressioni formate da una o più colonne presenti nelle tabelle a cui fa riferimento la query
- Numeri ordinali che rappresentano la posizione delle voci dell'elenco di selezione (o la posizione delle colonne nella tabella se non esiste alcun elenco di selezione)
- Alias che definiscono le voci dell'elenco di selezione

Quando la clausola ORDER BY contiene più espressioni, il set di risultati viene ordinato in base alla prima espressione, quindi la seconda espressione viene applicata alle righe che presentano valori corrispondenti della prima espressione e così via.

ASC | DESC

Opzione che definisce l'ordinamento per l'espressione, come segue:

- ASC: crescente (ad esempio, dal più piccolo al più grande per i valori numerici e da 'A' a 'Z' per le stringhe di caratteri). Se non viene specificata alcuna opzione, i dati vengono ordinati in ordine crescente per impostazione predefinita.

- DESC: decrescente (ad esempio, dal più grande al più piccolo per i valori numerici e da 'Z' ad 'A' per le stringhe).

NULLS FIRST | NULLS LAST

Opzione che specifica se i valori NULL devono essere ordinati per primi, prima dei valori non null, o per ultimi, dopo i valori non null. Per impostazione predefinita, i valori NULL vengono ordinati e classificati per ultimi in ordine ASC e ordinati e classificati per primi in ordine DESC.

LIMIT number | ALL

Opzione che controlla il numero di righe ordinate restituite dalla query. Il numero LIMIT deve essere un integer positivo; il valore massimo è 2147483647.

LIMIT 0 non restituisce righe. Puoi utilizzare questa sintassi a scopo di test: per verificare che una query venga eseguita (senza visualizzare alcuna riga) o per restituire un elenco di colonne da una tabella. Una clausola ORDER BY è ridondante se si utilizza LIMIT 0 per restituire un elenco di colonne. L'impostazione predefinita è LIMIT ALL.

OFFSET start

Opzione che specifica di ignorare il numero di righe prima di start prima di iniziare a restituire righe. Il numero OFFSET deve essere un integer intero positivo; il valore massimo è 2147483647. Se utilizzato con l'opzione LIMIT, le righe OFFSET vengono ignorate prima di iniziare a contare le righe LIMIT restituite. Se non si utilizza l'opzione LIMIT, il numero di righe nel set dei risultati viene ridotto del numero di righe che vengono ignorate. Le righe ignorate da una clausola OFFSET devono ancora essere analizzate, quindi potrebbe essere inefficiente utilizzare un valore OFFSET di grandi dimensioni.

Note per l'utilizzo

Nota il seguente comportamento previsto con le clausole ORDER BY:

- I valori NULL sono considerati "superiori" rispetto a tutti gli altri valori. Con l'ordine di ordinamento crescente predefinito, i valori NULL vengono ordinati alla fine. Per modificare questo comportamento, utilizza l'opzione NULLS FIRST.
- Quando una query non contiene una clausola ORDER BY, il sistema restituisce serie di risultati senza ordine prevedibile delle righe. La stessa query eseguita due volte potrebbe restituire il set di risultati in un ordine diverso.

- Le opzioni LIMIT e OFFSET possono essere utilizzate senza una clausola ORDER BY; tuttavia, per restituire un insieme coerente di righe, utilizza queste opzioni insieme a ORDER BY.
- In qualsiasi sistema parallelo come Amazon Redshift, quando ORDER BY non produce un ordinamento univoco, l'ordine delle righe è non deterministico. Ovvero, se l'espressione ORDER BY produce valori duplicati, l'ordine di restituzione di tali righe potrebbe variare da altri sistemi o da una sola esecuzione di Amazon Redshift a quella successiva.
- Amazon Redshift non supporta letterali stringa nelle clausole ORDER BY.

Esempi di ORDER BY

Restituisce tutte le 11 righe dalla tabella CATEGORY, ordinate in base alla seconda colonna CATGROUP. Per i risultati con lo stesso valore CATGROUP, ordina i valori della colonna CATDESC in base alla lunghezza della stringa di caratteri. Quindi ordina per colonne CATID e CATNAME.

```
select * from category order by 2, length(catdesc), 1, 3;

catid | catgroup |  catname  |                 catdesc
------+----------+-----------+--------------------------------------
10    | Concerts | Jazz      | All jazz singers and bands
9     | Concerts | Pop       | All rock and pop music concerts
11    | Concerts | Classical | All symphony, concerto, and choir conce
6     | Shows    | Musicals  | Musical theatre
7     | Shows    | Plays     | All non-musical theatre
8     | Shows    | Opera     | All opera and light opera
5     | Sports   | MLS       | Major League Soccer
1     | Sports   | MLB       | Major League Baseball
2     | Sports   | NHL       | National Hockey League
3     | Sports   | NFL       | National Football League
4     | Sports   | NBA       | National Basketball Association
(11 rows)
```

Restituisce le colonne selezionate dalla tabella SALES, ordinate in base ai valori QTYSOLD più alti. Limita il risultato alle prime 10 righe:

```
select salesid, qtysold, pricepaid, commission, saletime from sales
order by qtysold, pricepaid, commission, salesid, saletime desc
limit 10;

salesid | qtysold | pricepaid | commission |      saletime
--------+---------+-----------+------------+---------------------
```

```
15401   |       8 |    272.00 |      40.80 | 2008-03-18 06:54:56
61683   |       8 |    296.00 |      44.40 | 2008-11-26 04:00:23
90528   |       8 |    328.00 |      49.20 | 2008-06-11 02:38:09
74549   |       8 |    336.00 |      50.40 | 2008-01-19 12:01:21
130232  |       8 |    352.00 |      52.80 | 2008-05-02 05:52:31
55243   |       8 |    384.00 |      57.60 | 2008-07-12 02:19:53
16004   |       8 |    440.00 |      66.00 | 2008-11-04 07:22:31
489     |       8 |    496.00 |      74.40 | 2008-08-03 05:48:55
4197    |       8 |    512.00 |      76.80 | 2008-03-23 11:35:33
16929   |       8 |    568.00 |      85.20 | 2008-12-19 02:59:33
(10 rows)
```

Restituisce un elenco di colonne e nessuna riga usando la sintassi LIMIT 0:

```
select * from venue limit 0;
venueid | venuename | venuecity | venuestate | venueseats
---------+-----------+-----------+------------+------------
(0 rows)
```

Clausola CONNECT BY

La clausola CONNECT BY specifica la relazione tra le righe in una gerarchia. È possibile utilizzare CONNECT BY per selezionare le righe in ordine gerarchico eseguendo il joining della tabella con se stessa ed elaborando i dati gerarchici. Ad esempio, può essere usata per eseguire spostamenti in modo ricorsivo all'interno di un organigramma ed elencare i dati.

L'elaborazione delle query gerarchiche avviene nel seguente ordine:

1. Se la clausola FROM ha un join, viene elaborata per prima.
2. Viene valutata la clausola CONNECT BY.
3. Viene valutata la clausola WHERE.

Sintassi

```
[START WITH start_with_conditions]
CONNECT BY connect_by_conditions
```

> **Note**
>
> Sebbene START e CONNECT non siano parole riservate, utilizza identificatori delimitati (virgolette doppie) o AS se si utilizza START e CONNECT come alias di tabella nella query per evitare errori in fase di runtime.

```
SELECT COUNT(*)
FROM Employee "start"
CONNECT BY PRIOR id = manager_id
START WITH name = 'John'
```

```
SELECT COUNT(*)
FROM Employee AS start
CONNECT BY PRIOR id = manager_id
START WITH name = 'John'
```

Parametri

start_with_conditions

Condizioni che specificano la riga o le righe principali della gerarchia

connect_by_conditions

Condizioni che specificano la relazione tra le righe principali e le righe secondarie della gerarchia. Almeno una condizione deve essere qualificata con l'operatore unario utilizzato per fare riferimento alla riga principale.

```
PRIOR column = expression
-- or
expression > PRIOR column
```

Operatori

È possibile utilizzare i seguenti operatori in una query CONNECT BY.

LEVEL

Pseudocolonna che restituisce il livello di riga corrente nella gerarchia. Restituisce 1 per la riga principale, 2 per la riga secondaria e così via.

PRIOR

Operatore unario che valuta l'espressione per la riga principale della riga corrente nella gerarchia.

Esempi

L'esempio seguente è una query CONNECT BY che restituisce il numero di dipendenti che riportano direttamente o indirettamente a John (non più di 4 livelli).

```
SELECT id, name, manager_id
FROM employee
WHERE LEVEL < 4
START WITH name = 'John'
CONNECT BY PRIOR id = manager_id;
```

Di seguito è riportato il risultato della query.

```
id      name      manager_id
------+----------+--------------
  101     John        100
  102     Jorge       101
  103     Kwaku       101
  110     Liu         101
  201     Sofía       102
  106     Mateo       102
  110     Nikki       103
  104     Paulo       103
  105     Richard     103
  120     Saanvi      104
  200     Shirley     104
  205     Zhang       104
```

La definizione di tabella per questo esempio è la seguente:

```
CREATE TABLE employee (
   id INT,
```

```
   name VARCHAR(20),
   manager_id INT
   );
```

Di seguito sono riportate le righe inserite nella tabella.

```
INSERT INTO employee(id, name, manager_id)  VALUES
(100, 'Carlos', null),
(101, 'John', 100),
(102, 'Jorge', 101),
(103, 'Kwaku', 101),
(110, 'Liu', 101),
(106, 'Mateo', 102),
(110, 'Nikki', 103),
(104, 'Paulo', 103),
(105, 'Richard', 103),
(120, 'Saanvi', 104),
(200, 'Shirley', 104),
(201, 'Sofía', 102),
(205, 'Zhang', 104);
```

Di seguito è riportato un organigramma per il dipartimento di John.

Esempi di sottoquery

Gli esempi seguenti mostrano diversi modi in cui le sottoquery si adattano alle query SELECT. Per un altro esempio dell'uso delle sottoquery, consultare Esempi di JOIN.

Sottoquery dell'elenco SELECT

L'esempio seguente contiene una sottoquery nell'elenco SELECT. Questa sottoquery è scalar: restituisce solo una colonna e un valore, che viene ripetuto nel risultato per ogni riga restituita dalla query esterna. La query confronta il valore Q1SALES che la sottoquery calcola con i valori di vendita per due altri trimestri (2 e 3) nel 2008, come definito dalla query esterna.

```
select qtr, sum(pricepaid) as qtrsales,
(select sum(pricepaid)
from sales join date on sales.dateid=date.dateid
where qtr='1' and year=2008) as q1sales
from sales join date on sales.dateid=date.dateid
```

```
where qtr in('2','3') and year=2008
group by qtr
order by qtr;

qtr  |  qtrsales   |   q1sales
-------+-------------+-------------
2     | 30560050.00 | 24742065.00
3     | 31170237.00 | 24742065.00
(2 rows)
```

Sottoquery della clausola WHERE

L'esempio seguente contiene una sottoquery di tabella nella clausola WHERE. Questa sottoquery produce più righe. In questo caso, le righe contengono solo una colonna, ma le sottoquery di tabella possono contenere più colonne e righe, proprio come qualsiasi altra tabella.

La query trova i primi 10 venditori in termini di numero di biglietti venduti. L'elenco dei primi 10 è limitato dalla sottoquery che rimuove gli utenti che vivono in città dove ci sono le sedi dei biglietti. Questa query può essere scritta in diversi modi; ad esempio, la sottoquery potrebbe essere riscritta come un join all'interno della query principale.

```
select firstname, lastname, city, max(qtysold) as maxsold
from users join sales on users.userid=sales.sellerid
where users.city not in(select venuecity from venue)
group by firstname, lastname, city
order by maxsold desc, city desc
limit 10;

firstname | lastname  |      city      | maxsold
-----------+-----------+----------------+---------
Noah       | Guerrero | Worcester      |       8
Isadora    | Moss     | Winooski       |       8
Kieran     | Harrison | Westminster    |       8
Heidi      | Davis    | Warwick        |       8
Sara       | Anthony  | Waco           |       8
Bree       | Buck     | Valdez         |       8
Evangeline | Sampson  | Trenton        |       8
Kendall    | Keith    | Stillwater     |       8
Bertha     | Bishop   | Stevens Point  |       8
Patricia   | Anderson | South Portland |       8
(10 rows)
```

Sottoquery della clausola WITH

Per informazioni, consultare Clausola WITH.

Sottoquery correlate

L'esempio seguente contiene una sottoquery correlata nella clausola WHERE; questo tipo di sottoquery contiene una o più correlazioni tra le sue colonne e le colonne prodotte dalla query esterna. In questo caso, la correlazione è `where s.listid=l.listid`. Per ogni riga prodotta dalla query esterna, la sottoquery viene eseguita per qualificare o squalificare la riga.

```
select salesid, listid, sum(pricepaid) from sales s
where qtysold=
(select max(numtickets) from listing l
where s.listid=l.listid)
group by 1,2
order by 1,2
limit 5;

salesid | listid |   sum
--------+--------+----------
 27     |     28 | 111.00
 81     |    103 | 181.00
 142    |    149 | 240.00
 146    |    152 | 231.00
 194    |    210 | 144.00
(5 rows)
```

Modelli di sottoquery correlate non supportate

Il pianificatore di query utilizza un metodo di riscrittura delle query denominato decorrelazione delle sottoquery per ottimizzare diversi modelli di sottoquery correlate per l'esecuzione in un ambiente MPP. Alcuni tipi di sottoquery correlate seguono modelli che Amazon Redshift non può decorrelare e non supporta. Le query che contengono i seguenti riferimenti di correlazione restituiscono errori:

- Riferimenti di correlazione che ignorano un blocco di query, noti anche come "riferimenti di correlazione ignorati". Ad esempio, nella seguente query, il blocco contenente il riferimento di correlazione e il blocco ignorato sono collegati da un predicato NOT EXISTS:

  ```
  select event.eventname from event
  where not exists
  ```

```
(select * from listing
where not exists
(select * from sales where event.eventid=sales.eventid));
```

Il blocco ignorato in questo caso è la sottoquery rispetto alla tabella LISTING. Il riferimento di correlazione correla le tabelle EVENT e SALES.

- Riferimenti di correlazione di una sottoquery che fa parte di una clausola ON in una query esterna:

```
select * from category
left join event
on category.catid=event.catid and eventid =
(select max(eventid) from sales where sales.eventid=event.eventid);
```

La clausola ON contiene un riferimento di correlazione da SALES nella sottoquery a EVENT nella query esterna.

- Riferimenti di correlazione che rispettano i valori null a una tabella di sistema di Amazon Redshift. Ad esempio:

```
select attrelid
from stv_locks sl, pg_attribute
where sl.table_id=pg_attribute.attrelid and 1 not in
(select 1 from pg_opclass where sl.lock_owner = opcowner);
```

- Riferimenti di correlazione da una sottoquery contenente una funzione finestra.

```
select listid, qtysold
from sales s
where qtysold not in
(select sum(numtickets) over() from listing l where s.listid=l.listid);
```

- Riferimenti in una colonna GROUP BY ai risultati di una sottoquery correlata. Ad esempio:

```
select listing.listid,
(select count (sales.listid) from sales where sales.listid=listing.listid) as list
from listing
group by list, listing.listid;
```

- Riferimenti di correlazione da una sottoquery con una funzione di aggregazione e una clausola GROUP BY, connessi alla query esterna da un predicato IN. Questa restrizione non si applica alle funzioni di aggregazione MIN e MAX. Per esempio:

```
select * from listing where listid in
(select sum(qtysold)
from sales
where numtickets>4
group by salesid);
```

SELECT INTO

Seleziona le righe definite da qualsiasi query e le inserisce in una nuova tabella. Puoi specificare se creare una tabella permanente o temporanea.

Sintassi

```
[ WITH with_subquery [, ...] ]
SELECT
[ TOP number ] [ ALL | DISTINCT ]
* | expression [ AS output_name ] [, ...]
INTO [ TEMPORARY | TEMP ] [ TABLE ] new_table
[ FROM table_reference [, ...] ]
[ WHERE condition ]
[ GROUP BY expression [, ...] ]
[ HAVING condition [, ...] ]
[ { UNION | INTERSECT | { EXCEPT | MINUS } } [ ALL ] query ]
[ ORDER BY expression
[ ASC | DESC ]
[ LIMIT { number | ALL } ]
[ OFFSET start ]
```

Per dettagli sui parametri di questo comando, vedi SELECT.

Esempi

Seleziona tutte le righe dalla tabella EVENT e crea una tabella NEWEVENT:

```
select * into newevent from event;
```

Seleziona il risultato di una query di aggregazione in una tabella temporanea denominata PROFITS:

```
select username, lastname, sum(pricepaid-commission) as profit
```

```
into temp table profits
from sales, users
where sales.sellerid=users.userid
group by 1, 2
order by 3 desc;
```

SET

Imposta il valore di un parametro di configurazione del server. Utilizzare il comando SET per sostituire un'impostazione solo per la durata della sessione o della transazione corrente.

Utilizza il comando RESET per restituire un parametro al valore predefinito.

È possibile modificare i parametri di configurazione del server in diversi modi. Per ulteriori informazioni, consulta Modifica della configurazione del server.

Sintassi

```
SET { [ SESSION | LOCAL ]
{ SEED | parameter_name } { TO | = }
{ value | 'value' | DEFAULT } |
SEED TO value }
```

La seguente istruzione imposta il valore di una variabile di contesto di sessione.

```
SET { [ SESSION | LOCAL ]
variable_name { TO | = }
{ value | 'value'  }
```

Parametri

SESSION

Specifica che l'impostazione è valida per la sessione corrente. Valore predefinito.

variable_name

Specifica il nome della variabile di contesto impostata per la sessione.

La convenzione di denominazione è un nome in due parti separato da un punto, ad esempio identifier.identifier. È consentito un solo separatore di punti. Utilizzare un identificatore che segua

le regole standard degli identificatori per Amazon Redshift Per ulteriori informazioni, consulta Nomi e identificatori. Gli identificativi delimitati non sono ammessi.

LOCAL

Specifica che l'impostazione è valida per la transazione corrente.

SEED TO value

Imposta un seed interno che deve essere utilizzata dalla funzione RANDOM per la generazione di numeri casuali.

SET SEED accetta un valore numerico compreso tra 0 e 1 e moltiplica questo numero per (2^{31}-1) per l'uso con la funzione Funzione RANDOM. Se utilizzi SET SEED prima di effettuare più chiamate RANDOM, RANDOM genera numeri in una sequenza prevedibile.

parameter_name

Nome del parametro da impostare. Per informazioni sui parametri, vedi Modifica della configurazione del server.

value

Nuovo valore del parametro. Utilizza le virgolette singole per impostare il valore su una stringa specifica. Se si utilizza SET SEED, questo parametro contiene il valore SEED.

DEFAULT

Imposta il parametro sul valore predefinito.

Esempi

Modifica di un parametro per la sessione corrente

L'esempio seguente imposta il datestyle:

```
set datestyle to 'SQL,DMY';
```

Impostazione di un gruppo di query per la gestione dei carichi di lavoro

Se i gruppi di query sono elencati in una definizione di coda come parte della configurazione WLM del cluster, puoi impostare il parametro QUERY_GROUP su un nome di gruppo di query elencato. Le query successive vengono assegnate alla coda di query associata. L'impostazione di

QUERY_GROUP rimane valida per la durata della sessione o finché non viene rilevato un comando RESET QUERY_GROUP.

Questo esempio esegue due query come parte del gruppo di query 'priority', quindi reimposta il gruppo di query.

```
set query_group to 'priority';
select tbl, count(*)from stv_blocklist;
select query, elapsed, substring from svl_qlog order by query desc limit 5;
reset query_group;
```

Per ulteriori informazioni, consulta Implementazione della gestione del carico di lavoro.

Modifica lo spazio dei nomi di identità predefinito per la sessione

Un utente del database può impostare. `default_identity_namespace` Questo esempio mostra come utilizzare `SET SESSION` per sovrascrivere l'impostazione per la durata della sessione corrente e quindi mostrare il nuovo valore del provider di identità. Viene utilizzato più comunemente quando si utilizza un provider di identità con Redshift e IAM Identity Center. Per ulteriori informazioni sull'utilizzo di un provider di identità con Redshift, consulta Connect Redshift with IAM Identity Center per offrire agli utenti un'esperienza di single sign-on.

```
SET SESSION default_identity_namespace = 'MYCO';

SHOW default_identity_namespace;
```

Dopo aver eseguito il comando, puoi eseguire un'istruzione GRANT o un'istruzione CREATE come la seguente:

```
GRANT SELECT ON TABLE mytable TO alice;

GRANT UPDATE ON TABLE mytable TO salesrole;

CREATE USER bob password 'md50c983d1a624280812631c5389e60d48c';
```

In questo caso, l'effetto dell'impostazione dello spazio dei nomi di identità predefinito equivale a anteporre lo spazio dei nomi a ogni identità. In questo esempio, viene sostituito da. `alice` `MYCO:alice` Per ulteriori informazioni sulle impostazioni relative alla configurazione di Redshift con IAM Identity Center, ALTER SYSTEM consulta e. ALTER IDENTITY PROVIDER

Impostazione di un'etichetta per un gruppo di query

Il parametro QUERY_GROUP definisce un'etichetta per una o più query eseguite nella stessa sessione dopo un comando SET. A sua volta, questa etichetta viene registrata quando vengono eseguite le query e può essere utilizzata per limitare i risultati restituiti dalle tabelle di sistema STL_QUERY e STV_INFLIGHT e dalla vista SVL_QLOG.

```
show query_group;
query_group
-------------
unset
(1 row)

set query_group to '6 p.m.';


show query_group;
query_group
-------------
6 p.m.
(1 row)

select * from sales where salesid=500;
salesid | listid | sellerid | buyerid | eventid | dateid | ...
---------+--------+----------+---------+---------+--------+-----
500 |    504 |     3858 |    2123 |    5871 |   2052 | ...
(1 row)

reset query_group;

select query, trim(label) querygroup, pid, trim(querytxt) sql
from stl_query
where label ='6 p.m.';
query | querygroup |  pid  |                  sql
-------+------------+-------+----------------------------------------
57 | 6 p.m.     | 30711 | select * from sales where salesid=500;
(1 row)
```

Le etichette dei gruppi di query sono un meccanismo utile per isolare singole query o gruppi di query eseguite come parte degli script. Non è necessario identificare e tenere traccia delle query tramite i loro ID; puoi monitorarli con le loro etichette.

Impostazione di un valore di seed per la generazione di numeri casuali

L'esempio seguente utilizza l'opzione SEED con SET per far sì che la funzione RANDOM generi numeri in una sequenza prevedibile.

Innanzitutto, restituisce tre interi RANDOM senza prima impostare il valore SEED:

```
select cast (random() * 100 as int);
int4
------
6
(1 row)

select cast (random() * 100 as int);
int4
------
68
(1 row)

select cast (random() * 100 as int);
int4
------
56
(1 row)
```

Ora impostare il valore SEED su `.25`, e restituire altri tre numeri RANDOM:

```
set seed to .25;

select cast (random() * 100 as int);
int4
------
21
(1 row)

select cast (random() * 100 as int);
int4
------
79
(1 row)

select cast (random() * 100 as int);
int4
------
12
```

```
(1 row)
```

Infine, ripristinare il valore SEED su `.25`, e verificare che RANDOM restituisca gli stessi risultati delle tre chiamate precedenti:

```
set seed to .25;

select cast (random() * 100 as int);
int4
------
21
(1 row)

select cast (random() * 100 as int);
int4
------
79
(1 row)

select cast (random() * 100 as int);
int4
------
12
(1 row)
```

Nell'esempio seguente viene impostata una variabile di contesto personalizzata.

```
SET app_context.user_id TO 123;
SET app_context.user_id TO 'sample_variable_value';
```

SET SESSION AUTHORIZATION

Imposta il nome utente per la sessione corrente.

Puoi utilizzare il comando SET SESSION AUTHORIZATION, ad esempio, per verificare l'accesso al database eseguendo temporaneamente una sessione o una transazione come utente non privilegiato. Per eseguire questo comando, è necessario essere un utente con privilegi avanzati del database.

Sintassi

```
SET [ LOCAL ] SESSION AUTHORIZATION { user_name | DEFAULT }
```

Parametri

LOCAL

Specifica che l'impostazione è valida per la transazione corrente. L'omissione di questo parametro indica che l'impostazione è valida per la sessione corrente.

user_name

Nome dell'utente da impostare. Il nome utente può essere scritto come identificatore o stringa letterale.

DEFAULT

Imposta il nome utente della sessione sul valore predefinito.

Esempi

L'esempio seguente imposta il nome utente per la sessione corrente su `dwuser`:

```
SET SESSION AUTHORIZATION 'dwuser';
```

L'esempio seguente imposta il nome utente per la transazione corrente su `dwuser`:

```
SET LOCAL SESSION AUTHORIZATION 'dwuser';
```

L'esempio seguente imposta il nome utente per la sessione corrente sul nome utente predefinito:

```
SET SESSION AUTHORIZATION DEFAULT;
```

SET SESSION CHARACTERISTICS

Questo comando è obsoleto.

MOSTRA

Visualizza il valore corrente di un parametro di configurazione del server. Questo valore può essere specifico per la sessione corrente se è attivo un comando SET. Per un elenco dei parametri di configurazione, consultare Informazioni di riferimento sulla configurazione.

Sintassi

```
SHOW { parameter_name | ALL }
```

L'istruzione seguente mostra il valore corrente di una variabile di contesto di sessione. Se la variabile non esiste, Amazon Redshift genera un errore.

```
SHOW variable_name
```

Parametri

parameter_name

Visualizza il valore corrente del parametro specificato.

ALL

Visualizza i valori correnti di tutti i parametri.

variable_name

Visualizza il valore corrente della variabile specificata.

Esempi

L'esempio seguente visualizza il valore per il parametro del gruppo di query:

```
show query_group;

query_group

unset
(1 row)
```

L'esempio seguente mostra un elenco di tutti i parametri e i relativi valori:

```
show all;
name        |   setting
--------------------+--------------
datestyle          | ISO, MDY
extra_float_digits | 0
query_group        | unset
search_path        | $user,public
statement_timeout  | 0
```

L'esempio seguente mostra il valore corrente della variabile specificata.

```
SHOW app_context.user_id;
```

SHOW COLUMNS

Mostra un elenco di colonne in una tabella, insieme ad alcuni attributi delle colonne.

Ogni riga di output è costituita da un elenco separato da virgole di nome del database, nome dello schema, nome della tabella, nome della colonna, posizione ordinale, valore predefinito della colonna, è annullabile, tipo di dati, lunghezza massima del carattere, precisione numerica e commenti. Per ulteriori informazioni su questi attributi, consulta SVV_ALL_COLUMNS.

Se il comando SHOW COLUMNS restituisce più di 10.000 colonne, viene restituito un errore.

Sintassi

```
SHOW COLUMNS FROM TABLE database_name.schema_name.table_name [LIKE 'filter_pattern']
 [LIMIT row_limit ]
```

Parametri

database_name

Il nome del database che contiene le tabelle da elencare.

Per mostrare le tabelle in un formato AWS Glue Data Catalog, specifica (`awsdatacatalog`) come nome del database e assicurati che la configurazione del sistema `data_catalog_auto_mount` sia impostata su. `true` Per ulteriori informazioni, consulta ALTER SYSTEM.

schema_name

Il nome dello schema che contiene le tabelle da elencare.

Per mostrare AWS Glue Data Catalog le tabelle, fornite il nome del AWS Glue database come nome dello schema.

table_name

Il nome dello tabella che contiene le colonne da elencare.

filter_pattern

Un'espressione di caratteri UTF-8 valida con il modello da associare ai nomi della tabella. L'opzione LIKE esegue una corrispondenza con distinzione tra maiuscole e minuscole e supporta i seguenti metacaratteri che corrispondono ai modelli:

Metacaratteri	Descrizione
%	Abbina qualsiasi sequenza di zero o più caratteri.
_	Abbina qualsiasi carattere singolo.

Se il filter_pattern non contiene metacaratteri, allora il modello rappresenta solo la stringa stessa; in questo caso LIKE agisce come l'operatore di uguaglianza.

row_limit

Il numero massimo di righe da restituire. Il row_limit può essere compreso tra 0 e 10.000.

Esempi

L'esempio seguente mostra le colonne del database Amazon Redshift denominate `dev` che sono nello schema `public` e tabella `tb`.

```
SHOW COLUMNS FROM TABLE dev.public.tb;

 database_name | schema_name | table_name | column_name | ordinal_position
 | column_default | is_nullable | data_type | character_maximum_length |
 numeric_precision | remarks
```

```
--------------+------------+-----------+------------+-----------------
+---------------+------------+----------+-------------------------
+-------------------+---------
 dev           | public      | tb         | col         |               1 |
     | YES         | integer   |                          |               32 |
```

L'esempio seguente mostra le tabelle del AWS Glue Data Catalog database denominate `awsdatacatalog` che si trovano nello schema `batman` e nella tabella`nation`. L'output è limitato a 2 righe.

```
SHOW COLUMNS FROM TABLE awsdatacatalog.batman.nation LIMIT 2;

 database_name  | schema_name | table_name | column_name | ordinal_position
 | column_default | is_nullable | data_type | character_maximum_length |
 numeric_precision | remarks
---------------+------------+-----------+------------+-----------------
+---------------+------------+----------+-------------------------
+-------------------+---------
 awsdatacatalog | batman      | nation     | n_nationkey |               1 |
      |             | integer   |                          |                 |
 awsdatacatalog | batman      | nation     | n_name      |               2 |
      |             | character |                          |                 |
```

SHOW EXTERNAL TABLE

Mostra la definizione di una tabella esterna, inclusi gli attributi delle tabelle e gli attributi delle colonne. È possibile utilizzare l'output dell'istruzione SHOW EXTERNAL TABLE per ricreare la tabella.

Per ulteriori informazioni sulla creazione delle tabelle esterne, consultare CREATE EXTERNAL TABLE.

Sintassi

```
SHOW EXTERNAL TABLE [external_database].external_schema.table_name [ PARTITION ]
```

Parametri

external_database

Il nome del database esterno associato. Questo parametro è facoltativo.

external_schema

Il nome dello schema esterno associato.

table_name

Il nome della tabella da visualizzare.

PARTITION

Visualizza le istruzioni ALTER TABLE per aggiungere le partizioni alla definizione della tabella.

Esempi

Gli esempi seguenti si basano su una tabella esterna definita come segue:

```
CREATE EXTERNAL TABLE my_schema.alldatatypes_parquet_test_partitioned (
     csmallint smallint,
     cint int,
     cbigint bigint,
     cfloat float4,
     cdouble float8,
     cchar char(10),
     cvarchar varchar(255),
     cdecimal_small decimal(18,9),
     cdecimal_big decimal(30,15),
     ctimestamp TIMESTAMP,
     cboolean boolean,
     cstring varchar(16383)
)
PARTITIONED BY (cdate date, ctime TIMESTAMP)
STORED AS PARQUET
LOCATION 's3://mybucket-test-copy/alldatatypes_parquet_partitioned';
```

Di seguito è riportato un esempio del comando SHOW EXTERNAL TABLE e l'output per la tabella `my_schema.alldatatypes_parquet_test_partitioned`.

```
SHOW EXTERNAL TABLE my_schema.alldatatypes_parquet_test_partitioned;
```

```
"CREATE EXTERNAL TABLE my_schema.alldatatypes_parquet_test_partitioned (
    csmallint smallint,
```

```
    cint int,
    cbigint bigint,
    cfloat float4,
    cdouble float8,
    cchar char(10),
    cvarchar varchar(255),
    cdecimal_small decimal(18,9),
    cdecimal_big decimal(30,15),
    ctimestamp timestamp,
    cboolean boolean,
    cstring varchar(16383)
)
PARTITIONED BY (cdate date, ctime timestamp)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION 's3://mybucket-test-copy/alldatatypes_parquet_partitioned';"
```

Di seguito è riportato un esempio del comando SHOW EXTERNAL TABLE e dell'output per la stessa tabella, ma con il database specificato anche nel parametro.

```
SHOW EXTERNAL TABLE my_database.my_schema.alldatatypes_parquet_test_partitioned;
```

```
"CREATE EXTERNAL TABLE my_database.my_schema.alldatatypes_parquet_test_partitioned (
    csmallint smallint,
    cint int,
    cbigint bigint,
    cfloat float4,
    cdouble float8,
    cchar char(10),
    cvarchar varchar(255),
    cdecimal_small decimal(18,9),
    cdecimal_big decimal(30,15),
    ctimestamp timestamp,
    cboolean boolean,
    cstring varchar(16383)
)
PARTITIONED BY (cdate date, ctime timestamp)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION 's3://mybucket-test-copy/alldatatypes_parquet_partitioned';"
```

Di seguito è riportato un esempio del comando SHOW EXTERNAL TABLE e l'output quando si utilizza il parametro PARTITION. L'output contiene istruzioni ALTER TABLE per aggiungere le partizioni alla definizione della tabella.

```
SHOW EXTERNAL TABLE my_schema.alldatatypes_parquet_test_partitioned PARTITION;
```

```
"CREATE EXTERNAL TABLE my_schema.alldatatypes_parquet_test_partitioned (
    csmallint smallint,
    cint int,
    cbigint bigint,
    cfloat float4,
    cdouble float8,
    cchar char(10),
    cvarchar varchar(255),
    cdecimal_small decimal(18,9),
    cdecimal_big decimal(30,15),
    ctimestamp timestamp,
    cboolean boolean,
    cstring varchar(16383)
)
PARTITIONED BY (cdate date)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION 's3://mybucket-test-copy/alldatatypes_parquet_partitioned';
ALTER TABLE my_schema.alldatatypes_parquet_test_partitioned ADD IF NOT
 EXISTS PARTITION (cdate='2021-01-01') LOCATION 's3://mybucket-test-copy/
alldatatypes_parquet_partitioned2/cdate=2021-01-01';
ALTER TABLE my_schema.alldatatypes_parquet_test_partitioned ADD IF NOT
 EXISTS PARTITION (cdate='2021-01-02') LOCATION 's3://mybucket-test-copy/
alldatatypes_parquet_partitioned2/cdate=2021-01-02';"
```

SHOW DATABASES

Mostra i database a partire da un ID account specificato.

Sintassi

```
SHOW DATABASES FROM
DATA CATALOG [ ACCOUNT  '<id1>', '<id2>', ... ]
[ LIKE '<expression>' ]
```

```
[ IAM_ROLE default | 'SESSION' | 'arn:aws:iam::<account-id>:role/<role-name>' ]
```

Parametri

ACCOUNT '<id1>', '<id2>', ...

Gli AWS Glue Data Catalog account da cui elencare i database. L'omissione di questo parametro indica che Amazon Redshift deve mostrare i database dell'account proprietario del cluster.

LIKE '<expression>'

Filtra l'elenco di database mostrando solo quelli che soddisfano l'espressione specificata. Questo parametro supporta modelli che utilizzano i caratteri jolly % (percento) e _ (carattere di sottolineatura).

IAM_ROLE default | 'SESSION' | 'arn:aws:iam::<account-id>:role/<role-name>'

Se specifichi un ruolo IAM associato al cluster quando esegui il comando SHOW DATABASES, Amazon Redshift utilizzerà le credenziali del ruolo quando esegui le query sul database.

Specificare la parola chiave `default` significa utilizzare il ruolo IAM impostato come predefinito e associato al cluster.

Utilizza `'SESSION'` se ti connetti al cluster Amazon Redshift utilizzando un'identità federata e accedi alle tabelle dallo schema esterno creato con il comando the section called “CREATE DATABASE”. Per ulteriori informazioni, consulta l'argomento relativo a Utilizzo di un'identità federata per gestire l'accesso di Amazon Redshift alle risorse locali e alle tabelle esterne di Amazon Redshift Spectrum, che illustra come configurare l'identità federata.

L'Amazon Resource Name (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione. Come minimo, il ruolo IAM deve disporre dell'autorizzazione per eseguire un'operazione LIST sul bucket Amazon S3 a cui accedere e un'operazione GET sugli oggetti Amazon S3 contenuti nel bucket. Per ulteriori informazioni sui database creati da AWS Glue Data Catalog for datashare e utilizzando IAM_ROLE, vedi Lavorare con le condivisioni di dati gestite da Lake Formation come consumatore.

Quanto segue mostra la sintassi per la stringa di parametro IAM_ROLE per un singolo ARN.

```
IAM_ROLE 'arn:aws:iam::<aws-account-id>:role/<role-name>'
```

È possibile concatenare i ruoli in modo che il cluster possa presumere un altro ruolo IAM, possibilmente appartenente a un altro account. Puoi concatenare fino a 10 ruoli. Per ulteriori informazioni, consulta Concatenazione di ruoli IAM per Amazon Redshift Spectrum.

Per collegare a questo ruolo IAM una policy di autorizzazioni IAM simile alla seguente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessSecret",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-west-2:123456789012:secret:my-
rds-secret-VNenFy"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword",
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*"
        }
    ]
}
```

Per la procedura per creare un ruolo IAM da utilizzare con la query federata, consultare Creazione di un segreto e di un ruolo IAM per l'utilizzo di query federate.

> **Note**
>
> Non includere spazi nell'elenco dei ruoli concatenati.

Quanto segue mostra la sintassi per concatenare tre ruoli.

```
IAM_ROLE 'arn:aws:iam::<aws-account-id>:role/<role-1-name>,arn:aws:iam::<aws-
account-id>:role/<role-2-name>,arn:aws:iam::<aws-account-id>:role/<role-3-name>'
```

Esempi

L'esempio seguente mostra tutti i database del catalogo dati dell'account con ID 123456789012.

```
SHOW DATABASES FROM DATA CATALOG ACCOUNT '123456789012'

  catalog_id  | database_name |                          database_arn
 |     type     |                                          target_database
                            | location | parameters
--------------+---------------+-------------------------------------------------------------
+--------------
+-------------------------------------------------------------------------------------------
+----------+------------
 123456789012 |   database1   | arn:aws:glue:us-east-1:123456789012:database/database1
 | Data Catalog |
                            |          |
 123456789012 |   database2   | arn:aws:glue:us-east-1:123456789012:database/database2
 | Data Catalog | arn:aws:redshift:us-
east-1:123456789012:datashare:035c45ea-61ce-86f0-8b75-19ac6102c3b7/database2 |
 |
```

Di seguito sono riportati esempi che dimostrano come visualizzare tutti i database del Catalogo dati dall'account con ID 123456789012 utilizzando le credenziali di un ruolo IAM.

```
SHOW DATABASES FROM DATA CATALOG ACCOUNT '123456789012' IAM_ROLE default;
```

```
SHOW DATABASES FROM DATA CATALOG ACCOUNT '123456789012' IAM_ROLE <iam-role-arn>;
```

SHOW MODEL

Mostra le informazioni utili su un modello di machine learning, inclusi lo stato, i parametri utilizzati per crearlo e la funzione di previsione con i relativi tipi di argomenti di input. È possibile utilizzare le informazioni da SHOW MODEL per ricreare il modello. Se le tabelle di base sono state modificate,

l'esecuzione di CREATE MODEL con la stessa istruzione SQL genera un modello diverso. Le informazioni restituite da SHOW MODEL sono diverse per il proprietario del modello e per un utente con il privilegio EXECUTE. SHOW MODEL mostra diversi output quando un modello viene addestrato da Amazon Redshift o quando il modello è un modello BYOM.

Sintassi

```
SHOW MODEL ( ALL | model_name )
```

Parametri

ALL

Restituisce tutti i modelli che l'utente può utilizzare e i relativi schemi.

model_name

Il nome del modello. Il nome del modello in uno schema deve essere unico.

Note per l'utilizzo

Il comando SHOW MODEL restituisce:

- Il nome del modello.
- Lo schema in cui è stato creato il modello.
- Il proprietario del modello.
- L'ora di creazione del modello.
- Lo stato del modello, ad esempio READY, TRAINING o FAILED.
- Il messaggio del motivo per un modello fallito.
- L'errore di convalida se il modello ha terminato l'addestramento.
- Il costo stimato necessario per ricavare il modello per un approccio non BYOM. Solo il proprietario del modello può visualizzare queste informazioni.
- Un elenco dei parametri specificati dall'utente e dei relativi valori, in particolare i seguenti elementi:
  - La colonna TARGET specificata.
  - Il tipo di modello, AUTO o XGBoost.
  - Il tipo di problema, come REGRESSION, BINARY_CLASSIFICATION, MULTICLASS_CLASSIFICATION. Questo parametro è specifico di AUTO.

- Il nome del processo di SageMaker formazione di Amazon o del processo Amazon SageMaker Autopilot che ha creato il modello. Puoi utilizzare questo nome di lavoro per trovare ulteriori informazioni sul modello su Amazon SageMaker.
- L'obiettivo, come MSE, F1, Accuracy. Questo parametro è specifico di AUTO.
- Il nome della funzione creata.
- Il tipo di inferenza, locale o remota.
- Gli argomenti di input della funzione di previsione.
- I tipi di argomenti di input della funzione di previsione per i modelli che non sono BYOM (bring your own model).
- Il tipo restituito della funzione di previsione. Questo parametro è specifico di BYOM.
- Il nome dell' SageMaker endpoint Amazon per un modello BYOM con inferenza remota.
- Il ruolo IAM. Solo il proprietario del modello può vederlo.
- Il bucket S3 utilizzato. Solo il proprietario del modello può vederlo.
- La AWS KMS chiave, se ne è stata fornita una. Solo il proprietario del modello può vederlo.
- Il tempo massimo di esecuzione del modello.
- Se il tipo di modello non è AUTO, Amazon Redshift mostra anche l'elenco degli iperparametri forniti e i relativi valori.

È inoltre possibile visualizzare alcune delle informazioni fornite da SHOW MODEL in altre tabelle di catalogo, come pg_proc. Amazon Redshift restituisce informazioni sulla funzione di previsione registrata nella tabella del catalogo pg_proc. Queste informazioni includono i nomi degli argomenti di input e i relativi tipi per la funzione di previsione. Amazon Redshift restituisce le stesse informazioni nel comando SHOW MODEL.

```
SELECT * FROM pg_proc WHERE proname ILIKE '%<function_name>%';
```

Esempi

L'esempio seguente mostra l'output di show model.

```
SHOW MODEL ALL;

Schema Name |  Model Name
------------+---------------
 public     | customer_churn
```

Il proprietario di customer_churn può visualizzare il seguente output. Un utente con solo il privilegio EXECUTE non può visualizzare il ruolo IAM, il bucket Amazon S3 e il costo stimato della modalità.

```
SHOW MODEL customer_churn;

       Key                  |           Value
----------------------------+---------------------------------
 Model Name                 | customer_churn
 Schema Name                | public
 Owner                      | 'owner'
 Creation Time              | Sat, 15.01.2000 14:45:20
 Model State                | READY
 validation:F1              | 0.855
 Estimated Cost             | 5.7
                            |
 TRAINING DATA:             |
 Table                      | customer_data
 Target Column              | CHURN
                            |
 PARAMETERS:                |
 Model Type                 | auto
 Problem Type               | binary_classification
 Objective                  | f1
 Function Name              | predict_churn
 Function Parameters        | age zip average_daily_spend average_daily_cases
 Function Parameter Types   | int int float float
 IAM Role                   | 'iam_role'
 KMS Key                    | 'kms_key'
 Max Runtime                | 36000
```

SHOW DATASHARES

Visualizza le condivisioni in entrata e in uscita in un cluster dallo stesso account o tra account. Se non si specifica un nome di una unità di condivisione dati, Amazon Redshift visualizza tutte le unità di condivisione dati in tutti i database del cluster. Gli utenti che dispongono dei privilegi ALTER e SHARE possono visualizzare le condivisioni per le quali dispongono dei privilegi.

Sintassi

```
SHOW DATASHARES [ LIKE 'namepattern' ]
```

Parametri

LIKE

Una clausola facoltativa che confronta il modello di nome specificato con la descrizione dell'unità di condivisione dati. Quando si utilizza questa clausola, Amazon Redshift visualizza solo le unità di condivisione dati con nomi che corrispondono al modello di nome specificato.

namepattern

Il nome dell'unità di condivisione dati richiesta o parte del nome da associare utilizzando caratteri jolly.

Esempi

L'esempio seguente mostra le unità di condivisione dati in entrata e in uscita in un cluster.

```
SHOW DATASHARES;
SHOW DATASHARES LIKE 'sales%';

share_name   | share_owner | source_database | consumer_database | share_type |
 createdate          | is_publicaccessible | share_acl | producer_account |
 producer_namespace
-------------+-------------+-----------------+-------------------+------------
+---------------------+---------------------+-----------+------------------
+--------------------------------------
'salesshare' | 100         | dev             |                   | outbound
 | 2020-12-09 01:22:54.| False               |           |   123456789012   |
 13b8833d-17c6-4f16-8fe4-1a018f5ed00d
```

SHOW PROCEDURE

Mostra la definizione di una determinata procedura archiviata, inclusa la relativa firma. Puoi utilizzare l'output di un SHOW PROCEDURE per ricreare la procedura archiviata.

Sintassi

```
SHOW PROCEDURE sp_name [( [ [ argname ] [ argmode ] argtype [, ...] ] )]
```

Parametri

sp_name

Il nome della procedura da mostrare.

[argname] [argmode] argtype

Tipi di argomento di input per individuare la procedura archiviata. Facoltativamente, puoi includere i tipi di dati dell'argomento completi, inclusi gli argomenti OUT. Questa parte è facoltativa se il nome della procedura archiviata è univoco (ossia non in overload).

Esempi

L'esempio seguente mostra la definizione della procedura `test_spl2`.

```
show procedure test_sp2(int, varchar);
                                        Stored Procedure Definition
-------------------------------------------------------------------------------------------
CREATE OR REPLACE PROCEDURE public.test_sp2(f1 integer, INOUT f2 character varying, OUT
 character varying)
LANGUAGE plpgsql
AS $_$
DECLARE
out_var alias for $3;
loop_var int;
BEGIN
IF f1 is null OR f2 is null THEN
RAISE EXCEPTION 'input cannot be null';
END IF;
CREATE TEMP TABLE etl(a int, b varchar);
FOR loop_var IN 1..f1 LOOP
insert into etl values (loop_var, f2);
f2 := f2 || '+' || f2;
END LOOP;
SELECT INTO out_var count(*) from etl;
END;
$_$

(1 row)
```

SHOW SCHEMAS

Mostra un elenco di schemi in un database, insieme ad alcuni attributi dello schema.

Ogni riga di output è composta dal nome del database, dal nome dello schema, dal proprietario dello schema, dal tipo di schema, dall'ACL dello schema, dal database di origine e dall'opzione dello schema. Per ulteriori informazioni su questi attributi, consulta SVV_ALL_SCHEMAS.

Se dal comando SHOW SCHEMAS possono risultare più di 10.000 schemi, viene restituito un errore.

Sintassi

```
SHOW SCHEMAS FROM DATABASE database_name [LIKE 'filter_pattern'] [LIMIT row_limit ]
```

Parametri

database_name

Il nome del database che contiene le tabelle da elencare.

Per mostrare le tabelle in un file AWS Glue Data Catalog, specificate (`awsdatacatalog`) come nome del database e assicuratevi che la configurazione del sistema `data_catalog_auto_mount` sia impostata su`true`. Per ulteriori informazioni, consulta ALTER SYSTEM.

filter_pattern

Un'espressione di caratteri UTF-8 valida con il modello da associare ai nomi dello schema. L'opzione LIKE esegue una corrispondenza con distinzione tra maiuscole e minuscole e supporta i seguenti metacaratteri che corrispondono ai modelli:

Metacaratteri	Descrizione
%	Abbina qualsiasi sequenza di zero o più caratteri.
_	Abbina qualsiasi carattere singolo.

Se il filter_pattern non contiene metacaratteri, allora il modello rappresenta solo la stringa stessa; in questo caso LIKE agisce come l'operatore di uguaglianza.

row_limit

Il numero massimo di righe da restituire. Il row_limit può essere compreso tra 0 e 10.000.

Esempi

L'esempio seguente mostra gli schemi del database Amazon Redshift denominato dev.

```
SHOW SCHEMAS FROM DATABASE dev;

 database_name |     schema_name      | schema_owner | schema_type |         schema_acl
          | source_database | schema_option
---------------+----------------------+--------------+-------------
+-----------------------------+-----------------+--------------
 dev           | pg_automv            |            1 | local       |
          |                 |
 dev           | pg_catalog           |            1 | local       | jpuser=UC/
jpuser~=U/jpuser  |                 |
 dev           | public               |            1 | local       | jpuser=UC/
jpuser~=UC/jpuser |                 |
 dev           | information_schema   |            1 | local       | jpuser=UC/
jpuser~=U/jpuser  |                 |
 dev           | schemad79cd6d93bf043 |            1 | local       |
          |                 |
```

L'esempio seguente mostra gli schemi del AWS Glue Data Catalog database denominatoawsdatacatalog. Il numero massimo di righe di output è 5.

```
SHOW SCHEMAS FROM DATABASE awsdatacatalog LIMIT 5;

 database_name  |     schema_name      | schema_owner | schema_type | schema_acl |
 source_database | schema_option
----------------+----------------------+--------------+-------------+------------
+-----------------+--------------
 awsdatacatalog | 000_too_many_glue_db |              | EXTERNAL    |            |
           |
 awsdatacatalog | 123_default          |              | EXTERNAL    |            |
           |
 awsdatacatalog | adhoc                |              | EXTERNAL    |            |
           |
 awsdatacatalog | all_shapes_10mb      |              | EXTERNAL    |            |
           |
```

```
 awsdatacatalog | all_shapes_1g        |             | EXTERNAL    |           |
           |
```

SHOW TABLE

Mostra la definizione di una tabella, inclusi gli attributi di tabella, i vincoli di tabella, gli attributi di colonna e i vincoli di colonna. È possibile utilizzare l'output dell'istruzione SHOW TABLE per creare nuovamente la tabella.

Per ulteriori informazioni sulla creazione di una tabella, consultare CREATE TABLE.

Sintassi

```
SHOW TABLE [schema_name.]table_name
```

Parametri

schema_name

(Facoltativo) Il nome dello schema correlato.

table_name

Il nome della tabella da visualizzare.

Esempi

Di seguito è riportato un esempio del comando SHOW TABLE e l'output per la tabella `sales`.

```
show table sales;
```

```
CREATE TABLE public.sales (
salesid integer NOT NULL ENCODE az64,
listid integer NOT NULL ENCODE az64 distkey,
sellerid integer NOT NULL ENCODE az64,
buyerid integer NOT NULL ENCODE az64,
eventid integer NOT NULL ENCODE az64,
dateid smallint NOT NULL,
qtysold smallint NOT NULL ENCODE az64,
pricepaid numeric(8,2) ENCODE az64,
commission numeric(8,2) ENCODE az64,
saletime timestamp without time zone ENCODE az64
```

```
)
DISTSTYLE KEY SORTKEY ( dateid );
```

Di seguito è riportato un esempio dell'output del comando SHOW TABLE per la tabella `category` nello schema `public`.

```
show table public.category;
```

```
CREATE TABLE public.category (
catid smallint NOT NULL distkey,
catgroup character varying(10) ENCODE lzo,
catname character varying(10) ENCODE lzo,
catdesc character varying(50) ENCODE lzo
) DISTSTYLE KEY SORTKEY ( catid );
```

L'esempio seguente crea la tabella `foo` con una chiave primaria.

```
create table foo(a int PRIMARY KEY, b int);
```

I risultati SHOW TABLE visualizzano l'istruzione create con tutte le proprietà della tabella `foo`.

```
show table foo;
```

```
CREATE TABLE public.foo ( a integer NOT NULL ENCODE az64, b integer ENCODE az64,
 PRIMARY KEY (a) ) DISTSTYLE AUTO;
```

SHOW TABLES

Mostra un elenco di tabelle in uno schema, insieme ad alcuni attributi della tabella.

Ogni riga di output è composta dal nome del database, dal nome dello schema, dal nome della tabella, dal tipo di tabella, dall'ACL della tabella e dai commenti. Per ulteriori informazioni su questi attributi, consulta SVV_ALL_TABLES.

Se il comando SHOW TABLES restituisce più di 10.000 tabelle, viene restituito un errore.

Sintassi

```
SHOW TABLES FROM SCHEMA database_name.schema_name [LIKE 'filter_pattern']
 [LIMIT row_limit ]
```

Parametri

database_name

Il nome del database che contiene le tabelle da elencare.

Per mostrare le tabelle in un AWS Glue Data Catalog, specificate (`awsdatacatalog`) come nome del database e assicuratevi che la configurazione del sistema `data_catalog_auto_mount` sia impostata su. `true` Per ulteriori informazioni, consulta ALTER SYSTEM.

schema_name

Il nome dello schema che contiene le tabelle da elencare.

Per mostrare AWS Glue Data Catalog le tabelle, fornite il nome del AWS Glue database come nome dello schema.

filter_pattern

Un'espressione di caratteri UTF-8 valida con il modello da associare ai nomi della tabella. L'opzione LIKE esegue una corrispondenza con distinzione tra maiuscole e minuscole e supporta i seguenti metacaratteri che corrispondono ai modelli:

Metacaratteri	Descrizione
%	Abbina qualsiasi sequenza di zero o più caratteri.
_	Abbina qualsiasi carattere singolo.

Se il filter_pattern non contiene metacaratteri, allora il modello rappresenta solo la stringa stessa; in questo caso LIKE agisce come l'operatore di uguaglianza.

row_limit

Il numero massimo di righe da restituire. Il row_limit può essere compreso tra 0 e 10.000.

Esempi

L'esempio seguente mostra le tabelle nel database Amazon Redshift denominate `dev` che sono nello schema `public`.

```
SHOW TABLES FROM SCHEMA dev.public;

 database_name | schema_name | table_name | table_type | table_acl | remarks
---------------+-------------+------------+------------+-----------+---------
 dev           | public      | tb         | TABLE      |           |
 dev           | public      | tb2        | TABLE      |           |
 dev           | public      | tb3        | TABLE      |           |
```

L'esempio seguente mostra le tabelle del AWS Glue Data Catalog database denominate `awsdatacatalog` che fanno parte dello schema`batman`.

```
SHOW TABLES FROM SCHEMA awsdatacatalog.batman;

 database_name  | schema_name |    table_name    | table_type | table_acl | remarks
----------------+-------------+------------------+------------+-----------+---------
 awsdatacatalog | batman      | nation           | EXTERNAL   |           |
 awsdatacatalog | batman      | part             | EXTERNAL   |           |
 awsdatacatalog | batman      | partsupp         | EXTERNAL   |           |
 awsdatacatalog | batman      | region           | EXTERNAL   |           |
 awsdatacatalog | batman      | supplier         | EXTERNAL   |           |
 awsdatacatalog | batman      | automount_nation | EXTERNAL   |           |
```

SHOW VIEW

Mostra la definizione di una vista, incluse le viste materializzate e le viste con associazione tardiva. È possibile utilizzare l'output dell'istruzione SHOW VIEW per creare nuovamente la tabella.

Sintassi

```
SHOW VIEW [schema_name.]view_name
```

Parametri

schema_name

(Facoltativo) Il nome dello schema correlato.

view_name

Il nome della vista da mostrare.

Esempi

Di seguito è riportata la definizione della vista per la vista LA_Venues_v.

```
create view LA_Venues_v as select * from venue where venuecity='Los Angeles';
```

Di seguito è riportato un esempio del comando SHOW VIEW e l'output per la vista definita in precedenza.

```
show view LA_Venues_v;
```

```
SELECT venue.venueid,
venue.venuename,
venue.venuecity,
venue.venuestate,
venue.venueseats
FROM venue WHERE ((venue.venuecity)::text = 'Los Angeles'::text);
```

Di seguito è riportata la definizione della vista public.Sports_v nello schema public.

```
create view public.Sports_v as select * from category where catgroup='Sports';
```

Di seguito è riportato un esempio del comando SHOW VIEW e l'output per la vista definita in precedenza.

```
show view public.Sports_v;
```

```
SELECT category.catid,
category.catgroup,
category.catname,
category.catdesc
FROM category WHERE ((category.catgroup)::text = 'Sports'::text);
```

START TRANSACTION

Sinonimo della funzione BEGIN.

Per informazioni, consultare BEGIN.

TRUNCATE

Elimina tutte le righe da una tabella senza eseguire una scansione della tabella: questa operazione è un'alternativa più rapida a un'operazione DELETE non qualificata. Per eseguire un comando TRUNCATE, è necessario disporre dell'autorizzazione TRUNCATE TABLE, essere il proprietario della tabella o un superutente. Per concedere le autorizzazioni per troncare una tabella, utilizza il comando GRANT.

TRUNCATE è molto più efficiente di DELETE e non richiede VACUUM e ANALYZE. Tuttavia, tieni presente che TRUNCATE esegue il commit della transazione in cui viene eseguito.

Sintassi

```
TRUNCATE [ TABLE ] table_name
```

Il comando funziona anche su una vista materializzata.

```
TRUNCATE materialized_view_name
```

Parametri

TABLE

Parola chiave facoltativa.

table_name

Tabella temporanea o persistente. Solo il proprietario della tabella o un utente con privilegi avanzati può troncarla.

Puoi troncare qualsiasi tabella, incluse le tabelle a cui si fa riferimento nei vincoli di chiave esterna.

Non è necessario sottoporre a vacuum una tabella dopo averla troncata.

materialized_view_name

Una vista materializzata.

È possibile troncare una vista materializzata utilizzata per Importazione di dati in streaming.

Note per l'utilizzo

Il comando TRUNCATE esegue il commit della transazione in cui viene eseguito; di conseguenza, non è possibile eseguire il rollback di un'operazione TRUNCATE e un comando TRUNCATE può eseguire altre operazioni quando esegue il commit.

Esempi

Utilizza il comando TRUNCATE per eliminare tutte le righe dalla tabella CATEGORY:

```
truncate category;
```

Tenta di eseguire il rollback di un'operazione TRUNCATE:

```
begin;

truncate date;

rollback;

select count(*) from date;
count
-------
0
(1 row)
```

La tabella DATE rimane vuota dopo il comando ROLLBACK perché il comando TRUNCATE è stato eseguito automaticamente.

L'esempio seguente utilizza il comando TRUNCATE per eliminare tutte le righe da una vista materializzata.

```
truncate my_materialized_view;
```

Elimina tutti i record nella vista materializzata e lascia intatti la vista materializzata e il relativo schema. Nella query, il nome della vista materializzata è un esempio.

UNLOAD

Scarica il risultato di una query in uno o più file di testo, JSON o Apache Parquet in Amazon S3 tramite la crittografia lato server di Amazon S3 (SSE-S3). Puoi anche specificare la crittografia lato

server con una chiave AWS Key Management Service (SSE-KMS) oppure la crittografia lato client con una chiave gestita dal cliente (CSE-CMK).

Per impostazione predefinita, il formato del file scaricato è testo delimitato da pipe (|).

È possibile gestire la dimensione dei file su Amazon S3 e, per estensione, il numero di file, impostando il parametro MAXFILESIZE. Assicurati che gli intervalli IP S3 siano aggiunti all'elenco di indirizzi consentiti. Per ulteriori informazioni sugli intervalli IP S3 richiesti, consulta Isolamento di rete.

È possibile scaricare il risultato di una query di Amazon Redshift sul proprio data lake Amazon S3 in Apache Parquet, un formato di archiviazione a colonne aperte per l'analisi particolarmente efficiente. Il formato Parquet è fino a 2 volte più veloce a scaricare e consuma fino a 6 volte meno spazio di archiviazione in Amazon S3 rispetto ai formati di testo. Ciò consente di salvare la trasformazione e l'arricchimento dei dati eseguita in Amazon S3 nel data lake Amazon S3 in un formato aperto. Puoi quindi analizzare i tuoi dati con Redshift Spectrum e altri AWS servizi come Amazon Athena, Amazon EMR e Amazon. SageMaker

Per ulteriori informazioni e scenari di esempio sull'utilizzo del comando UNLOAD, consulta Scaricamento dei dati.

Privilegi e autorizzazioni richiesti

Affinché il comando UNLOAD abbia esito positivo, è necessario almeno il privilegio SELECT sui dati nel database, insieme all'autorizzazione per scrivere nella posizione Amazon S3. Le autorizzazioni necessarie sono simili al comando COPY. Per ulteriori informazioni sulla gestione delle autorizzazioni, consulta Autorizzazioni per accedere ad altre risorse AWS.

Sintassi

```
UNLOAD ('select-statement')
TO 's3://object-path/name-prefix'
authorization
[ option, ...]

where authorization is
IAM_ROLE { default | 'arn:aws:iam::<Account AWS-id-1>:role/<role-
name>[,arn:aws:iam::<Account AWS-id-2>:role/<role-name>][,...]' }

where option is
| [ FORMAT [ AS ] ] CSV | PARQUET | JSON
```

```
| PARTITION BY ( column_name [, ... ] ) [ INCLUDE ]
| MANIFEST [ VERBOSE ]
| HEADER
| DELIMITER [ AS ] 'delimiter-char'
| FIXEDWIDTH [ AS ] 'fixedwidth-spec'
| ENCRYPTED [ AUTO ]
| BZIP2
| GZIP
| ZSTD
| ADDQUOTES
| NULL [ AS ] 'null-string'
| ESCAPE
| ALLOWOVERWRITE
| CLEANPATH
| PARALLEL [ { ON | TRUE } | { OFF | FALSE } ]
| MAXFILESIZE [AS] max-size [ MB | GB ]
| ROWGROUPSIZE [AS] size [ MB | GB ]
| REGION [AS] 'aws-region' }
| EXTENSION 'extension-name'
```

Parametri

('select-statement')

Query SELECT. I risultati della query vengono scaricati. Nella maggior parte dei casi, vale la pena scaricare i dati nell'ordine ordinato specificando una clausola ORDER BY nella query. Questo approccio risparmia il tempo richiesto per ordinare i dati quando vengono ricaricati.

La query deve essere racchiusa tra virgolette singole, come mostrato di seguito:

```
('select * from venue order by venueid')
```

> **Note**
>
> Se la query contiene virgolette (ad esempio per racchiudere valori letterali), inserire il valore letterale tra due set di virgolette singole. È inoltre necessario racchiudere la query tra virgolette singole:
>
> ```
> ('select * from venue where venuestate=''NV''')
> ```

TO 's3://object-path/name-prefix'

Il percorso completo, incluso il nome del bucket, nella posizione su Amazon S3 dove Amazon Redshift scrive gli oggetti del file di output, incluso il file manifest se è specificato MANIFEST. Ai nomi degli oggetti viene anteposto name-prefix. Se si utilizza `PARTITION BY`, viene aggiunta automaticamente una barra alla fine del valore name-prefix, se necessario. Per maggiore sicurezza, UNLOAD si collega ad Amazon S3 utilizzando una connessione HTTPS. Per impostazione predefinita, UNLOAD scrive uno o più file per sezione. UNLOAD aggiunge un numero di sezioni e un numero di parte al prefisso del nome specificato come segue:

<object-path>/*<name-prefix><slice-number>*_part_*<part-number>*.

Se viene specificato MANIFEST, il file manifest viene scritto come segue:

<object_path>/*<name_prefix>*manifest.

Se PARALLEL è specificato OFF, i file di dati vengono scritti come segue:

<object_path>/*<name_prefix><part-number>*.

UNLOAD crea automaticamente file crittografati usando la crittografia lato server (SSE) Amazon S3, incluso il file manifest se viene utilizzato MANIFEST. Il comando COPY legge automaticamente i file crittografati sul lato server durante l'operazione di caricamento. È possibile scaricare in modo trasparente i file crittografati sul lato server dal bucket utilizzando la console di Amazon S3 o l'API. Per ulteriori informazioni, consultare Protezione dei dati con la crittografia lato server.

Per usare la crittografia lato client di Amazon S3, specificare l'opzione ENCRYPTED.

> **Important**
>
> REGION è obbligatorio quando il bucket Amazon S3 non si trova nella stessa Regione AWS del database Amazon Redshift.

authorization

Il comando UNLOAD necessita l'autorizzazione per scrivere dati in Amazon S3. Il comando UNLOAD utilizza gli stessi parametri usati dal comando COPY per l'autorizzazione. Per ulteriori

informazioni, consultare Parametri di autorizzazione nel riferimento della sintassi del comando COPY.

IAM_ROLE { default | 'arn:aws:iam::*<Account AWS-id-1>*:role/*<role-name>*'

Utilizzare la parola chiave predefinita per fare in modo che Amazon Redshift utilizzi il ruolo IAM impostato come predefinito e associato al cluster quando viene eseguito il comando UNLOAD.

L'Amazon Resource Name (ARN) per un ruolo IAM utilizzato dal cluster per l'autenticazione e l'autorizzazione. Se specifichi IAM_ROLE, non è possibile utilizzare ACCESS_KEY_ID e SECRET_ACCESS_KEY, SESSION_TOKEN o CREDENTIALS. IAM_ROLE può essere concatenato. Per ulteriori informazioni, consulta Concatenazione di ruoli IAM nella Guida alla gestione di Amazon Redshift.

[FORMAT [AS]] CSV | PARQUET | JSON

Le parole chiave per specificare il formato di scarico che sovrascrive il formato predefinito.

Scarica in un file di testo in formato CSV utilizzando una virgola (,) come delimitatore predefinito. Se un campo contiene delimitatori, virgolette doppie, caratteri newline o ritorni a capo, il campo nel file scaricato è racchiuso tra virgolette doppie. A una virgoletta doppia all'interno di un campo dati viene aggiunta un'ulteriore virgoletta doppia. Quando vengono scaricate zero righe, Amazon Redshift potrebbe scrivere oggetti Amazon S3 vuoti.

Nel caso di PARQUET, viene scaricata in un file in formato Apache Parquet versione 1.0. Per impostazione predefinita, ogni gruppo di righe viene compresso mediante la compressione SNAPPY. Per ulteriori informazioni sul formato Apache Parquet, consultare Parquet.

Nel caso di JSON, scarica in un file JSON con ogni riga contenente un oggetto JSON, che rappresenta un record completo nel risultato della query. Amazon Redshift supporta la scrittura di JSON nidificati quando il risultato della query contiene colonne SUPER. Per creare un oggetto JSON valido, il nome di ogni colonna della query deve essere univoco. Nel file JSON, i valori booleani vengono scaricati come `t` o `f` e i valori NULL vengono scaricati come `null`. Quando vengono scaricate zero righe, Amazon Redshift non scrive oggetti Amazon S3 vuoti.

Le parole chiave FORMAT e AS sono facoltative. Non è possibile utilizzare CSV con FIXEDWIDTH o ADDQUOTES. Non è possibile utilizzare PARQUET con DELIMITER, FIXEDWIDTH, ADDQUOTES, ESCAPE, NULL AS, HEADER, GZIP, BZIP2 o ZSTD. PARQUET con ENCRYPTED è supportato solo con la crittografia lato server con una AWS Key Management Service chiave (SSE-KMS). Non è possibile utilizzare JSON con DELIMITER, HEADER, FIXEDWIDTH, ADDQUOTES, ESCAPE o NULL AS.

PARTITION BY (column_name [, ...]) [INCLUDE]

Specifica le chiavi di partizione per l'operazione di scaricamento. UNLOAD suddivide automaticamente i file di output in cartelle di partizione in base ai valori chiave di partizione, secondo la convenzione Apache Hive. Ad esempio, un file Parquet che appartiene all'anno di partizione 2019 e al mese di settembre ha il seguente prefisso: `s3://my_bucket_name/my_prefix/year=2019/month=September/000.parquet`.

Il valore per column_name deve essere una colonna nei risultati della query che vengono scaricati.

Se si specifica PARTITION BY con l'opzione INCLUDE, le colonne delle partizioni non vengono rimosse dai file scaricati.

Amazon Redshift non supporta letterali stringa nelle clausole PARTITION BY.

MANIFEST [VERBOSE]

Crea un file manifest che elenca esplicitamente i dettagli per dati creati dal processo UNLOAD. Il manifest è un file di testo in formato JSON che elenca l'URL di ciascun file scritto in Amazon S3.

Se viene specificato MANIFEST con l'opzione VERBOSE, il manifest include i seguenti dettagli:

- I nomi della colonna e i tipi di dati e per i tipi di dati CHAR, VARCHAR o NUMERIC le dimensioni di ciascuna colonna. Per i tipi di dati CHAR e VARCHAR, la dimensione è la lunghezza. Per un tipo di dati DECIMAL o NUMERIC, le dimensioni sono la precisione e la scalabilità.
- Numero di righe scaricate in ciascun file. Se è specificata l'opzione HEADER, il numero di righe include la riga dell'intestazione.
- Le dimensioni totali del file di tutti i file scaricati e il numero totale di righe scaricate su tutti i file. Se è specificata l'opzione HEADER, il numero di righe include le righe dell'intestazione.
- L'autore. L'autore è sempre "Amazon Redshift".

Puoi specificare VERBOSE solo dopo MANIFEST.

Il file manifest è scritto nello stesso prefisso del percorso Amazon S3 come file scaricati nel formato `<object_path_prefix>manifest`. Ad esempio, se UNLOAD specifica il prefisso del percorso di Amazon S3 '`s3://mybucket/venue_`', il percorso del file manifest sarà '`s3://mybucket/venue_manifest`'.

HEADER

Aggiunge una riga di intestazione che contiene i nomi delle colonne nella parte superiore di ogni file output. Anche le opzioni di trasformazione del testo, come CSV, DELIMITER, ADDQUOTES ed ESCAPE, si applicano alla riga di intestazione. Non è possibile utilizzare HEADER con FIXEDWIDTH.

DELIMITER AS 'delimiter_character'

Specifica un singolo carattere ASCII utilizzato per separare i campi del file di output, ad esempio un carattere pipe (|), una virgola (,) o una tabulazione (\t). Il delimitatore predefinito per i file di testo è un carattere pipe. Il delimitatore predefinito per i file CSV è una virgola. La parola chiave AS è facoltativa. Non è possibile utilizzare DELIMITER con FIXEDWIDTH. Se i dati contengono il carattere delimitatore, è necessario specificare l'opzione ESCAPE per evitare il delimitatore o utilizzare ADDQUOTES per racchiudere i dati tra virgolette doppie. In alternativa, specifica un delimitatore non contenuto nei dati.

FIXEDWIDTH 'fixedwidth_spec'

Scarica i dati in un file in cui ogni larghezza di colonna è fissa, invece di essere separate da un delimitatore. La stringa fixedwidth_spec specifica il numero di colonne e la larghezza delle colonne. La parola chiave AS è facoltativa. Poiché FIXEDWIDTH non tronca i dati, la specifica per ogni colonna nell'istruzione UNLOAD deve essere lunga almeno quanto la lunghezza della voce più lunga per quella colonna. Di seguito è indicato il formato di fixedwidth_spec:

```
'colID1:colWidth1,colID2:colWidth2, ...'
```

Non è possibile utilizzare FIXEDWIDTH con DELIMITER o HEADER.

ENCRYPTED [AUTO]

Specifica che i file di output in Amazon S3 saranno crittografati usando la crittografia lato server o la crittografia lato client di Amazon S3. Se viene specificato MANIFEST, anche il file manifest viene crittografato. Per ulteriori informazioni, consulta Scaricamento di file di dati crittografati. Se non specifichi il parametro ENCRYPTED, UNLOAD crea automaticamente file crittografati utilizzando la crittografia lato server di Amazon S3 AWS con chiavi di crittografia gestite (SSE-S3).

Per ENCRYPTED, potresti voler scaricare su Amazon S3 utilizzando la crittografia lato server con AWS KMS una chiave (SSE-KMS). In tal caso, utilizza il parametro KMS_KEY_ID per fornire l'ID della chiave. Non puoi usare il parametro CREDENTIALS con il parametro KMS_KEY_ID. Se

esegui un comando UNLOAD per i dati utilizzando KMS_KEY_ID, puoi eseguire un'operazione COPY per gli stessi dati senza specificare una chiave.

Per scaricare in Amazon S3 utilizzando la crittografia lato client con una chiave simmetrica fornita dal cliente fornisce la chiave in due modi. Per fornire la chiave, utilizza il parametro MASTER_SYMMETRIC_KEY o la parte `master_symmetric_key` di una stringa di credenziali CREDENTIALS. Se scarichi i dati utilizzando una chiave simmetrica master, assicurati di fornire la stessa chiave quando esegui un'operazione COPY per i dati crittografati.

UNLOAD non supporta la crittografia lato server di Amazon S3 con una chiave fornita dal cliente (SSE-C).

Se viene utilizzato ENCRYPTED AUTO, il comando UNLOAD recupera la chiave di AWS KMS crittografia predefinita sulla proprietà del bucket Amazon S3 di destinazione e crittografa i file scritti su Amazon S3 con la chiave. AWS KMS Se il bucket non dispone della chiave di AWS KMS crittografia predefinita, UNLOAD crea automaticamente file crittografati utilizzando la crittografia lato server di Amazon Redshift AWS con chiavi di crittografia gestite (SSE-S3). Non è possibile utilizzare questa opzione con KMS_KEY_ID, MASTER_SYMMETRIC_KEY o CREDENTIALS contenente master_symmetric_key.

KMS_KEY_ID 'key-id'

Speciifica l'ID della chiave per una chiave AWS Key Management Service (AWS KMS) da utilizzare per crittografare i file di dati su Amazon S3. Per ulteriori informazioni, consulta What is? AWS Key Management Service Se specifichi KMS_KEY_ID, devi specificare anche il parametro ENCRYPTED. Se specifichi KMS_KEY_ID, non puoi autenticare usando il parametro CREDENTIALS. Utilizza invece IAM_ROLE o ACCESS_KEY_ID and SECRET_ACCESS_KEY.

MASTER_SYMMETRIC_KEY 'root_key'

La chiave simmetrica master da utilizzare per crittografare i file di dati in Amazon S3. Se specifichi il parametro MASTER_SYMMETRIC_KEY, devi anche specificare il parametro ENCRYPTED. Non è possibile utilizzare MASTER_SYMMETRIC_KEY con il parametro CREDENTIALS. Per ulteriori informazioni, consulta Caricamento di file di dati crittografati da Amazon S3.

BZIP2

Scarica i dati in uno o più file compressi con bzip2 per sezione. A ogni file risultante viene aggiunta l'estensione `.bz2`.

GZIP

Scarica i dati in uno o più file compressi con gzip per sezione. A ogni file risultante viene aggiunta l'estensione `.gz`.

ZSTD

Scarica i dati in uno o più file compressi con Zstandard per sezione. A ogni file risultante viene aggiunta l'estensione `.zst`.

ADDQUOTES

Mettere le virgolette attorno a ciascun campo dati scaricato, in modo che Amazon Redshift possa scaricare i valori dei dati che contengono il delimitatore stesso. Ad esempio, se il delimitatore è una virgola, puoi scaricare e ricaricare i seguenti dati:

```
"1","Hello, World"
```

Senza le virgolette aggiunte, la stringa `Hello, World` verrebbe analizzata come due campi separati.

Alcuni formati di output non supportano ADDQUOTES.

Se usi ADDQUOTES, devi specificare REMOVEQUOTES nel parametro COPY se ricarichi i dati.

NULL AS 'null-string'

Specifica una stringa che rappresenta un valore null nei file scaricato. Se si utilizza questa opzione, tutti i file di output contengono la stringa specificata al posto di qualsiasi valore null trovato nei dati selezionati. Se questa opzione non è specificata, i valori null vengono scaricati come:

- Stringhe a lunghezza zero per output delimitati
- Stringhe di spazi vuoti per output a larghezza fissa

Se viene specificata una stringa null per uno scaricamento a larghezza fissa e la larghezza di una colonna di output è inferiore alla larghezza della stringa null, si verifica il seguente comportamento:

- Un campo vuoto è l'output per le colonne non di caratteri
- Viene restituito un errore per le colonne di caratteri

A differenza di altri tipi di dati in cui una stringa definita dall'utente rappresenta un valore nullo, Amazon Redshift esporta le colonne di dati SUPER utilizzando il formato JSON e li rappresenta

come null come determinato dal formato JSON. Di conseguenza, le colonne di dati SUPER ignorano l'opzione NULL [AS] utilizzata nei comandi UNLOAD.

ESCAPE

Per le colonne CHAR e VARCHAR in file scaricati delimitati, un carattere di escape (\) viene inserito prima di ogni occorrenza dei seguenti caratteri:

- Avanzamento riga: `\n`
- Ritorno a capo: `\r`
- Il carattere delimitatore specificato per i dati scaricati.
- Il carattere di escape: `\`
- Il carattere virgoletta: `''` o `'` (se nel comando UNLOAD sono specificati entrambi, ESCAPE e ADDQUOTES).

> **Important**
>
> Se hai caricato i dati utilizzando un comando COPY con l'opzione ESCAPE, è necessario specificare anche l'opzione ESCAPE con il comando UNLOAD per generare il file di output reciproco. Allo stesso modo, se utilizzi UNLOAD usando l'opzione ESCAPE, è necessario usare ESCAPE quando effettui un COPY sugli stessi dati.

ALLOWOVERWRITE

Per impostazione predefinita, UNLOAD ha esito negativo se trova i file che potrebbe sovrascrivere. Se è specificato ALLOWOVERWRITE, UNLOAD sovrascrive i file esistenti, incluso il file manifest.

CLEANPATH

L'opzione CLEANPATH rimuove i file esistenti che si trovano nel percorso Amazon S3 specificato nella clausola TO prima di scaricare i file nella posizione specificata.

Se si include la clausola PARTITION BY, i file esistenti vengono rimossi solo dalle cartelle delle partizioni per ricevere i nuovi file generati dall'operazione UNLOAD.

È necessario disporre dell'autorizzazione `s3:DeleteObject` sul bucket Amazon S3. Per ulteriori informazioni, consultare Policy e autorizzazioni in Amazon S3 nella Guida per l'utente di Amazon

Simple Storage Service. I file rimossi utilizzando l'opzione CLEANPATH vengono eliminati in modo definitivo e non possono essere recuperati.

Non è possibile specificare l'opzione CLEANPATH se si specifica l'opzione ALLOWOVERWRITE.

PARALLEL

Per impostazione predefinita, UNLOAD scrive i dati in parallelo in più file, in base al numero di sezioni nel cluster. L'opzione predefinita è ON o TRUE. Se PARALLEL è OFF o FALSE, UNLOAD scrive in uno o più file di dati in serie, ordinati in modo assoluto in base alla clausola ORDER BY, se utilizzata. Le dimensioni massime per un file di dati sono di 6,2 GB. Ad esempio, se scarichi 13,4 GB di dati, UNLOAD crea i seguenti tre file.

```
s3://mybucket/key000    6.2 GB
s3://mybucket/key001    6.2 GB
s3://mybucket/key002    1.0 GB
```

> **Note**
>
> Il comando UNLOAD è progettato per usare l'elaborazione parallela. Nella maggior parte dei casi, è consigliabile lasciare abilitata l'opzione PARALLEL, in particolare se i file vengono usati per caricare le tabelle usando un comando COPY.

MAXFILESIZE [AS] max-size [MB | GB]

Specificare la dimensione massima dei file creati da UNLOAD in Amazon S3. Specifica un valore decimale compreso tra 5 MB e 6,2 GB. La parola chiave AS è facoltativa. L'unità predefinita è MB. Se MAXFILESIZE non è specificato, la dimensione massima del file predefinita è 6,2 GB. MAXFILESIZE non influisce sulle dimensioni del file manifest, se utilizzato.

ROWGROUPSIZE [AS] [MB | GB]

Specifica le dimensioni dei gruppi di righe. La scelta di dimensioni maggiori può ridurre il numero di gruppi di righe, riducendo la quantità di comunicazioni di rete. Specificare un valore intero compreso tra 32 MB e 128 MB. La parola chiave AS è facoltativa. L'unità predefinita è MB.

Se non è specificato ROWGROUPSIZE, la dimensione predefinita è 32 MB. Per utilizzare questo parametro, il formato di archiviazione deve essere Parquet e il tipo di nodo deve essere ra3.4xlarge, ra3.16xlarge o dc2.8xlarge.

REGION [AS] 'aws-region'

Speciifica Regione AWS dove si trova il bucket Amazon S3 di destinazione. REGION è necessario per UNLOAD su un bucket Amazon S3 che non si trova nello Regione AWS stesso database Amazon Redshift.

Il valore per aws_region deve corrispondere a una AWS regione elencata nella tabella delle regioni e degli endpoint di Amazon Redshift nel. Riferimenti generali di AWS

Per impostazione predefinita, UNLOAD presuppone che il bucket Amazon S3 di destinazione si trovi nello stesso database Amazon Regione AWS Redshift.

EXTENSION 'extension-name'

Specifica l'estensione del file da aggiungere ai nomi dei file scaricati. Amazon Redshift non esegue alcuna convalida, quindi è necessario verificare che l'estensione del file specificata sia corretta. Se stai usando un metodo di compressione come GZIP, devi comunque specificare `.gz` nel parametro dell'estensione. Se non fornisci alcuna estensione, Amazon Redshift non aggiunge nulla al nome del file. Se specifichi un metodo di compressione senza fornire un'estensione, Amazon Redshift aggiunge solo l'estensione del metodo di compressione al nome del file.

Note per l'utilizzo

Utilizzo di ESCAPE per tutte le operazioni UNLOAD di testo delimitato

Quando specifichi UNLOAD usando un delimitatore, i dati possono includere il delimitatore o uno qualsiasi dei caratteri elencati nella descrizione dell'opzione ESCAPE. In questo caso, devi usare l'opzione ESCAPE con l'istruzione UNLOAD. Se non usi l'opzione ESCAPE con UNLOAD, le successive operazioni COPY che utilizzano i dati scaricati potrebbero non riuscire.

> **Important**
>
> Si consiglia di utilizzare sempre ESCAPE sia con l'istruzione UNLOAD sia con quella COPY. A meno che tu non sia sicuro che i tuoi dati non contengano delimitatori o altri caratteri per cui si potrebbe dover utilizzare il carattere di escape.

Perdita della precisione a virgola mobile

Potresti riscontrare la perdita di precisione per i dati a virgola mobile che vengono successivamente scaricati e ricaricati.

Clausola Limit

La query SELECT non può utilizzare una clausola LIMIT nell'istruzione SELECT esterna. Ad esempio, la seguente istruzione UNLOAD non riesce.

```
unload ('select * from venue limit 10')
to 's3://mybucket/venue_pipe_' iam_role 'arn:aws:iam::0123456789012:role/
MyRedshiftRole';
```

Utilizza invece una clausola LIMIT nidificata, come nell'esempio seguente.

```
unload ('select * from venue where venueid in
(select venueid from venue order by venueid desc limit 10)')
to 's3://mybucket/venue_pipe_' iam_role 'arn:aws:iam::0123456789012:role/
MyRedshiftRole';
```

In alternativa, è possibile popolare una tabella usando SELECT…INTO o CREATE TABLE AS usando una clausola LIMIT, quindi scaricare da quella tabella.

Scaricare una colonna definita come tipo dati GEOMETRY

È possibile scaricare colonne di tipo GEOMETRY in formato testo o CSV. Non è possibile scaricare dati di tipo GEOMETRY con l'opzione `FIXEDWIDTH`. I dati vengono scaricati nella forma esadecimale del formato Extended Well-Known Binary (EWKB). Se la dimensione dei dati in formato EWKB è superiore a 4 MB, l'utente riceve l'avviso del fatto che i dati non possono essere successivamente caricati in una tabella.

Scarico del tipo di dati HLLSKETCH

È possibile scaricare colonne di tipo HLLSKETCH solo in formato testo o CSV. Non è possibile scaricare dati di tipo HLLSKETCH con l'opzione `FIXEDWIDTH`. I dati vengono scaricati nel formato Base64 per schizzi densi HyperLogLog o nel formato JSON per schizzi sparsi. HyperLogLog Per ulteriori informazioni, consulta HyperLogLog funzioni.

Nell'esempio seguente viene esportata una tabella contenente colonne HLLSKETCH in un file.

```
CREATE TABLE a_table(an_int INT, b_int INT);
INSERT INTO a_table VALUES (1,1), (2,1), (3,1), (4,1), (1,2), (2,2), (3,2), (4,2),
 (5,2), (6,2);

CREATE TABLE hll_table (sketch HLLSKETCH);
```

```
INSERT INTO hll_table select hll_create_sketch(an_int) from a_table group by b_int;

UNLOAD ('select * from hll_table') TO 's3://mybucket/unload/'
IAM_ROLE 'arn:aws:iam::0123456789012:role/MyRedshiftRole' NULL AS 'null' ALLOWOVERWRITE
 CSV;
```

Scaricare una colonna definita come tipo dati VARBYTE

È possibile scaricare colonne di tipo VARBYTE in formato testo o CSV. I dati vengono scaricati nel formato esadecimale. Non è possibile scaricare i dati VARBYTE con l'opzione `FIXEDWIDTH`. L'opzione `ADDQUOTES` di UNLOAD in un CSV non è supportata. Una colonna VARBYTE non può essere utilizzata come colonna di partizione.

Clausola FORMAT AS PARQUET

Tieni presenti queste considerazioni quando utilizzi FORMAT AS PARQUET:

- Lo scaricamento in Parquet non utilizza la compressione a livello di file. Ogni gruppo di righe viene compresso con SNAPPY.
- Se MAXFILESIZE non è specificato, la dimensione massima del file predefinita è 6,2 GB. È possibile utilizzare MAXFILESIZE per specificare una dimensione file di 5 MB-6,2 GB. La dimensione effettiva del file è approssimata quando il file viene scritto, quindi potrebbe non corrispondere esattamente al numero specificato.

  Per massimizzare le prestazioni di scansione, Amazon Redshift prova a creare file Parquet contenenti gruppi di righe da 32 MB di dimensioni uguali. Il valore MAXFILESIZE specificato viene arrotondato automaticamente al multiplo più vicino di 32 MB. Ad esempio, se si specifica 200 MB per MAXFILESIZE, ogni file Parquet scaricato è di circa 192 MB (gruppo di righe da 32 MB x 6 = 192 MB).
- Se una colonna utilizza il formato di dati TIMESTAMPTZ, vengono scaricati solo i valori di timestamp. Le informazioni sul fuso orario non vengono scaricate.
- Non specificare i prefissi dei nomi file che iniziano con caratteri di sottolineatura (_) o punto (.). Redshift Spectrum tratta i file che iniziano con questi caratteri come file nascosti e li ignora.

Clausola PARTITION BY

Tieni presenti queste considerazioni quando utilizzi PARTITION BY:

- Le colonne delle partizioni non sono incluse nel file di output.

- Assicurati di includere colonne di partizione nella query SELECT utilizzata nell'istruzione UNLOAD. È possibile specificare un numero qualsiasi di colonne di partizione nel comando UNLOAD. Esiste tuttavia una limitazione che prevede che almeno una colonna non di partizione faccia parte del file.
- Se il valore della chiave di partizione è null, Amazon Redshift scarica automaticamente i dati in una partizione di default chiamata `partition_column=__HIVE_DEFAULT_PARTITION__`.
- Il comando UNLOAD non effettua chiamate a un catalogo esterno. Per registrare le nuove partizioni in modo che facciano parte della tabella esterna esistente, utilizza un ALTER TABLE separato... Comando ADD PARTITION... Oppure è possibile eseguire un comando CREATE EXTERNAL TABLE per registrare i dati scaricati come una nuova tabella esterna. Puoi anche utilizzare un crawler per popolare il tuo Data Catalog AWS Glue . Per ulteriori informazioni, consultare Definizione di crawler nella Guida per gli sviluppatori di AWS Glue .
- Se si utilizza l'opzione MANIFEST, Amazon Redshift genera un solo file manifest nella cartella Amazon S3 principale.
- I tipi di dati di colonna che è possibile utilizzare come chiave di partizione sono SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, BOOLEAN, CHAR, VARCHAR, DATE e TIMESTAMP.

Utilizzo del privilegio ASSUMEROLE per concedere l'accesso a un ruolo IAM per le operazioni UNLOAD

Per fornire l'accesso per utenti e gruppi specifici a un ruolo IAM per le operazioni UNLOAD, un utente con privilegi avanzati può concedere il privilegio ASSUMEROLE su un ruolo IAM a utenti e gruppi. Per informazioni, consulta GRANT.

UNLOAD non supporta gli alias degli Access Point Amazon S3

È possibile utilizzare gli alias degli Access Point Amazon S3 con il comando UNLOAD.

Esempi

Per alcuni esempi che illustrano come utilizzare il comando UNLOAD, consulta Esempi di UNLOAD.

Esempi di UNLOAD

Questi esempi mostrano vari parametri del comando UNLOAD. La maggior parte degli esempi usa il set di dati TICKIT di esempio. Per ulteriori informazioni, consulta Database di esempio.

> **Note**
>
> Questi esempi contengono interruzioni di riga per una migliore leggibilità. Non includere interruzioni di riga o spazi nella stringa credentials-args.

Scarica VENUE in un file delimitato da pipe (delimitatore predefinito)

L'esempio seguente scarica la tabella VENUE e scrive i dati in `s3://mybucket/unload/`:

```
unload ('select * from venue')
to 's3://mybucket/unload/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Per impostazione predefinita, UNLOAD scrive uno o più file per sezione. Supponendo un cluster a due nodi con due sezioni per nodo, l'esempio precedente crea questi file in `mybucket`:

```
unload/0000_part_00
unload/0001_part_00
unload/0002_part_00
unload/0003_part_00
```

Per differenziare meglio i file di output, puoi includere un prefisso nella posizione. L'esempio seguente scarica la tabella VENUE e scrive i dati in `s3://mybucket/unload/venue_pipe_`:

```
unload ('select * from venue')
to 's3://mybucket/unload/venue_pipe_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Il risultato sono questi quattro file nella cartella `unload`, sempre presupponendo quattro sezioni.

```
venue_pipe_0000_part_00
venue_pipe_0001_part_00
venue_pipe_0002_part_00
venue_pipe_0003_part_00
```

Scaricare la tabella LINEITEM nei file Parquet partizionati

Nell'esempio seguente viene scaricata la tabella LINEITEM in formato Parquet, partizionata dalla colonna `l_shipdate`.

```
unload ('select * from lineitem')
to 's3://mybucket/lineitem/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
PARQUET
PARTITION BY (l_shipdate);
```

Supponendo quattro sezioni, i file Parquet risultanti vengono partizionati dinamicamente in varie cartelle.

```
s3://mybucket/lineitem/l_shipdate=1992-01-02/0000_part_00.parquet
                                             0001_part_00.parquet
                                             0002_part_00.parquet
                                             0003_part_00.parquet
s3://mybucket/lineitem/l_shipdate=1992-01-03/0000_part_00.parquet
                                             0001_part_00.parquet
                                             0002_part_00.parquet
                                             0003_part_00.parquet
s3://mybucket/lineitem/l_shipdate=1992-01-04/0000_part_00.parquet
                                             0001_part_00.parquet
                                             0002_part_00.parquet
                                             0003_part_00.parquet
...
```

Note

In alcuni casi, il comando UNLOAD viene utilizzato con l'opzione INCLUDE come illustrato nella seguente istruzione SQL.

```
unload ('select * from lineitem')
to 's3://mybucket/lineitem/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
PARQUET
PARTITION BY (l_shipdate) INCLUDE;
```

In questi casi, la colonna `l_shipdate` è presente anche nei dati nei file Parquet. In caso contrario, i dati delle colonne `l_shipdate` non si trovano nei file Parquet.

Scarica la tabella VENUE in un file JSON

L'esempio seguente scarica la tabella VENUE e scrive i dati in formato JSON in `s3://mybucket/unload/`.

```
unload ('select * from venue')
to 's3://mybucket/unload/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
JSON;
```

Di seguito sono riportare righe di esempio dalla tabella VENUE.

```
venueid | venuename                  | venuecity       | venuestate | venueseats
--------+----------------------------+-----------------+------------+-----------
      1 | Pinewood Racetrack         | Akron           | OH         | 0
      2 | Columbus "Crew" Stadium    | Columbus        | OH         | 0
      4 | Community, Ballpark, Arena | Kansas City     | KS         | 0
```

Dopo lo scaricamento in JSON, il formato del file è simile al seguente.

```
{"venueid":1,"venuename":"Pinewood
 Racetrack","venuecity":"Akron","venuestate":"OH","venueseats":0}
{"venueid":2,"venuename":"Columbus \"Crew\" Stadium
 ","venuecity":"Columbus","venuestate":"OH","venueseats":0}
{"venueid":4,"venuename":"Community, Ballpark, Arena","venuecity":"Kansas
 City","venuestate":"KS","venueseats":0}
```

Scarica VENUE in un file CSV

L'esempio seguente scarica la tabella VENUE e scrive i dati in formato CSV in `s3://mybucket/unload/`.

```
unload ('select * from venue')
to 's3://mybucket/unload/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
CSV;
```

Supponiamo che la tabella VENUE contenga le seguenti righe.

```
venueid | venuename                  | venuecity       | venuestate | venueseats
```

```
--------+----------------------------+-----------------+------------+-----------
      1 | Pinewood Racetrack         | Akron           | OH         | 0
      2 | Columbus "Crew" Stadium    | Columbus        | OH         | 0
      4 | Community, Ballpark, Arena | Kansas City     | KS         | 0
```

Il file di unload è simile al seguente.

```
1,Pinewood Racetrack,Akron,OH,0
2,"Columbus ""Crew"" Stadium",Columbus,OH,0
4,"Community, Ballpark, Arena",Kansas City,KS,0
```

Scarica VENUE in un file CSV utilizzando un delimitatore

L'esempio seguente scarica la tabella VENUE e scrive i dati in formato CSV utilizzando il carattere pipe (|) come delimitatore. Il file scaricato viene scritto su `s3://mybucket/unload/`. La tabella VENUE in questo esempio contiene il carattere pipe nel valore della prima riga (`Pinewood Race|track`). Lo fa per mostrare che il valore nel risultato è racchiuso tra virgolette doppie. Una virgoletta doppia è sottoposta a escape mediante una virgoletta doppia e l'intero campo è racchiuso tra virgolette doppie.

```
unload ('select * from venue')
to 's3://mybucket/unload/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
CSV DELIMITER AS '|';
```

Supponiamo che la tabella VENUE contenga le seguenti righe.

```
venueid | venuename                  | venuecity       | venuestate | venueseats
--------+----------------------------+-----------------+------------+-------------
      1 | Pinewood Race|track        | Akron           | OH         | 0
      2 | Columbus "Crew" Stadium    | Columbus        | OH         | 0
      4 | Community, Ballpark, Arena | Kansas City     | KS         | 0
```

Il file di unload è simile al seguente.

```
1|"Pinewood Race|track"|Akron|OH|0
2|"Columbus ""Crew"" Stadium"|Columbus|OH|0
4|Community, Ballpark, Arena|Kansas City|KS|0
```

Scarica VENUE con un file manifest

Per creare un file manifest, includi l'opzione MANIFEST. L'esempio seguente scarica la tabella VENUE e scrive un file manifest insieme ai file di dati in s3://mybucket/venue_pipe_:

> **Important**
>
> Se scarichi i file con l'opzione MANIFEST, devi utilizzare l'opzione MANIFEST con il comando COPY quando carichi i file. Se usi lo stesso prefisso per caricare i file e non specifichi l'opzione MANIFEST, COPY non riesce perché presuppone che il file manifest sia un file di dati.

```
unload ('select * from venue')
to 's3://mybucket/venue_pipe_' iam_role 'arn:aws:iam::0123456789012:role/
MyRedshiftRole'
manifest;
```

Il risultato sono questi cinque file:

```
s3://mybucket/venue_pipe_0000_part_00
s3://mybucket/venue_pipe_0001_part_00
s3://mybucket/venue_pipe_0002_part_00
s3://mybucket/venue_pipe_0003_part_00
s3://mybucket/venue_pipe_manifest
```

Di seguito viene mostrato il contenuto di un file manifest.

```
{
  "entries": [
    {"url":"s3://mybucket/tickit/venue_0000_part_00"},
    {"url":"s3://mybucket/tickit/venue_0001_part_00"},
    {"url":"s3://mybucket/tickit/venue_0002_part_00"},
    {"url":"s3://mybucket/tickit/venue_0003_part_00"}
  ]
}
```

Scarica VENUE con MANIFEST VERBOSE

Se viene specificata l'opzione MANIFEST VERBOSE, il file manifest include le seguenti sezioni:

- La sezione `entries` elenca il percorso Amazon S3, la dimensione del file e il conteggio righe per ciascun file.
- La sezione `schema` elenca i nomi della colonna, i tipi di dati e la dimensione di ciascuna colonna.
- La sezione `meta` mostra la dimensione totale del file e il conteggio righe per tutti i file.

L'esempio seguente scarica la tabella VENUE utilizzando l'opzione MANIFEST VERBOSE.

```
unload ('select * from venue')
to 's3://mybucket/unload_venue_folder/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
manifest verbose;
```

Di seguito viene mostrato il contenuto di un file manifest.

```
{
  "entries": [
    {"url":"s3://mybucket/venue_pipe_0000_part_00", "meta": { "content_length": 32295,
 "record_count": 10 }},
    {"url":"s3://mybucket/venue_pipe_0001_part_00", "meta": { "content_length": 32771,
 "record_count": 20 }},
    {"url":"s3://mybucket/venue_pipe_0002_part_00", "meta": { "content_length": 32302,
 "record_count": 10 }},
    {"url":"s3://mybucket/venue_pipe_0003_part_00", "meta": { "content_length": 31810,
 "record_count": 15 }}
  ],
  "schema": {
    "elements": [
      {"name": "venueid", "type": { "base": "integer" }},
      {"name": "venuename", "type": { "base": "character varying", 25 }},
      {"name": "venuecity", "type": { "base": "character varying", 25 }},
      {"name": "venuestate", "type": { "base": "character varying", 25 }},
      {"name": "venueseats", "type": { "base": "character varying", 25 }}
    ]
  },
  "meta": {
    "content_length": 129178,
    "record_count": 55
  },
  "author": {
    "name": "Amazon Redshift",
    "version": "1.0.0"
```

```
  }
}
```

Scarica VENUE con un Header

Di seguito è riportato un esempio di scaricamento di VENUE con una riga intestazione.

```
unload ('select * from venue where venueseats > 75000')
to 's3://mybucket/unload/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
header
parallel off;
```

Di seguito viene mostrato il contenuto di un file output con una riga intestazione.

```
venueid|venuename|venuecity|venuestate|venueseats
6|New York Giants Stadium|East Rutherford|NJ|80242
78|INVESCO Field|Denver|CO|76125
83|FedExField|Landover|MD|91704
79|Arrowhead Stadium|Kansas City|MO|79451
```

Scarica VENUE in file più piccoli

Le dimensioni massime predefinite per un file sono di 6,2 GB. Se i dati scaricati sono superiori a 6,2 GB, UNLOAD crea un nuovo file per ogni segmento di dati da 6,2 GB. Per creare file più piccoli, specifica il parametro MAXFILESIZE. Presupponendo che la dimensione dei dati nell'esempio precedente fosse 20 GB, il seguente comando UNLOAD crea 20 file, ciascuno di 1 GB di dimensione.

```
unload ('select * from venue')
to 's3://mybucket/unload/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
maxfilesize 1 gb;
```

Scarica VENUE in serie

Per scaricare in serie, specifica PARALLEL OFF. UNLOAD scrive quindi un file alla volta, fino a un massimo di 6,2 GB per file.

L'esempio seguente scarica la tabella VENUE e scrive i dati in serie in `s3://mybucket/unload/`.

```
unload ('select * from venue')
```

```
to 's3://mybucket/unload/venue_serial_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
parallel off;
```

Il risultato è un file chiamato venue_serial_000.

Se i dati scaricati sono superiori a 6,2 GB, UNLOAD crea un nuovo file per ogni segmento di dati da 6,2 GB. L'esempio seguente scarica la tabella LINEORDER e scrive i dati in serie in `s3://mybucket/unload/`.

```
unload ('select * from lineorder')
to 's3://mybucket/unload/lineorder_serial_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
parallel off gzip;
```

Il risultato è la seguente serie di file.

```
lineorder_serial_0000.gz
lineorder_serial_0001.gz
lineorder_serial_0002.gz
lineorder_serial_0003.gz
```

Per differenziare meglio i file di output, puoi includere un prefisso nella posizione. L'esempio seguente scarica la tabella VENUE e scrive i dati in `s3://mybucket/venue_pipe_`:

```
unload ('select * from venue')
to 's3://mybucket/unload/venue_pipe_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Il risultato sono questi quattro file nella cartella `unload`, sempre presupponendo quattro sezioni.

```
venue_pipe_0000_part_00
venue_pipe_0001_part_00
venue_pipe_0002_part_00
venue_pipe_0003_part_00
```

Carica VENUE da file scaricati

Per caricare una tabella da un set di file scaricati, è sufficiente invertire il processo utilizzando un comando COPY. L'esempio seguente crea una nuova tabella LOADVENUE e carica la tabella dai file di dati creati nell'esempio precedente.

```
create table loadvenue (like venue);

copy loadvenue from 's3://mybucket/venue_pipe_' iam_role
 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

Se hai usato l'opzione MANIFEST per creare un file manifest con i file scaricati, puoi caricare i dati utilizzando lo stesso file manifest. A questo scopo, puoi eseguire un comando COPY con l'opzione MANIFEST. L'esempio seguente carica i dati utilizzando un file manifest.

```
copy loadvenue
from 's3://mybucket/venue_pipe_manifest' iam_role 'arn:aws:iam::0123456789012:role/
MyRedshiftRole'
manifest;
```

Scarica VENUE in file crittografati

L'esempio seguente scarica la tabella VENUE in un set di file crittografati utilizzando una chiave. AWS KMS Se specifichi un file manifest con l'opzione ENCRYPTED, anche il file manifest viene crittografato. Per ulteriori informazioni, consulta Scaricamento di file di dati crittografati.

```
unload ('select * from venue')
to 's3://mybucket/venue_encrypt_kms'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
kms_key_id '1234abcd-12ab-34cd-56ef-1234567890ab'
manifest
encrypted;
```

L'esempio seguente scarica la tabella VENUE in un set di file crittografati utilizzando una chiave simmetrica master.

```
unload ('select * from venue')
to 's3://mybucket/venue_encrypt_cmk'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
master_symmetric_key 'EXAMPLEMASTERKEYtkbjk/OpCwtYSx/M4/t7DMCDIK722'
encrypted;
```

Carica VENUE da file crittografati

Per caricare le tabelle da un set di file creati usando UNLOAD con l'opzione ENCRYPT, devi invertire il processo utilizzando un comando COPY. Con quel comando, utilizza l'opzione ENCRYPTED e

specifica la stessa chiave simmetrica master utilizzata per il comando UNLOAD. L'esempio seguente carica la tabella LOADVENUE dai file di dati crittografati creati nell'esempio precedente.

```
create table loadvenue (like venue);

copy loadvenue
from 's3://mybucket/venue_encrypt_manifest'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
master_symmetric_key 'EXAMPLEMASTERKEYtkbjk/OpCwtYSx/M4/t7DMCDIK722'
manifest
encrypted;
```

Scarica dati di VENUE in un file delimitato da tabulazioni

```
unload ('select venueid, venuename, venueseats from venue')
to 's3://mybucket/venue_tab_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter as '\t';
```

I file di dati di output hanno l'aspetto seguente:

```
1 Toyota Park Bridgeview IL 0
2 Columbus Crew Stadium Columbus OH 0
3 RFK Stadium Washington DC 0
4 CommunityAmerica Ballpark Kansas City KS 0
5 Gillette Stadium Foxborough MA 68756
...
```

Scarica VENUE in un file di dati a larghezza fissa

```
unload ('select * from venue')
to 's3://mybucket/venue_fw_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
fixedwidth as 'venueid:3,venuename:39,venuecity:16,venuestate:2,venueseats:6';
```

I file di dati di output sono simili a quanto segue.

```
1  Toyota Park              Bridgeview  IL0
2  Columbus Crew Stadium    Columbus    OH0
3  RFK Stadium              Washington  DC0
4  CommunityAmerica BallparkKansas City KS0
```

```
5  Gillette Stadium         Foxborough  MA68756
...
```

Scarica VENUE in un set di file delimitati da tabulazioni e compressi con GZIP

```
unload ('select * from venue')
to 's3://mybucket/venue_tab_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter as '\t'
gzip;
```

Scaricamento di VENUE in un file di testo compresso con GZIP

```
unload ('select * from venue')
to 's3://mybucket/venue_tab_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
extension 'txt.gz'
gzip;
```

Scarica dati contenenti un delimitatore

Questo esempio utilizza l'opzione ADDQUOTES per scaricare i dati delimitati da virgole in cui alcuni dei campi di dati effettivi contengono una virgola.

Per prima cosa, crea una tabella che contenga virgolette.

```
create table location (id int, location char(64));

insert into location values (1,'Phoenix, AZ'),(2,'San Diego, CA'),(3,'Chicago, IL');
```

Quindi, scarica i dati usando l'opzione ADDQUOTES.

```
unload ('select id, location from location')
to 's3://mybucket/location_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
delimiter ',' addquotes;
```

I file di dati scaricati hanno l'aspetto seguente:

```
1,"Phoenix, AZ"
2,"San Diego, CA"
3,"Chicago, IL"
```

```
...
```

Scarica i risultati di una query di join

L'esempio seguente scarica i risultati di una query di join che contiene una funzione finestra.

```
unload ('select venuecity, venuestate, caldate, pricepaid,
sum(pricepaid) over(partition by venuecity, venuestate
order by caldate rows between 3 preceding and 3 following) as winsum
from sales join date on sales.dateid=date.dateid
join event on event.eventid=sales.eventid
join venue on event.venueid=venue.venueid
order by 1,2')
to 's3://mybucket/tickit/winsum'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

I file di output hanno l'aspetto seguente:

```
Atlanta|GA|2008-01-04|363.00|1362.00
Atlanta|GA|2008-01-05|233.00|2030.00
Atlanta|GA|2008-01-06|310.00|3135.00
Atlanta|GA|2008-01-08|166.00|8338.00
Atlanta|GA|2008-01-11|268.00|7630.00
...
```

Scarica utilizzando NULL AS

UNLOAD restituisce valori null come stringhe vuote per impostazione predefinita. I seguenti esempi mostrano come usare NULL AS per sostituire una stringa di testo per null.

Per questi esempi, aggiungiamo valori null alla tabella VENUE.

```
update venue set venuestate = NULL
where venuecity = 'Cleveland';
```

Seleziona da VENUE dove VENUESTATE è null per verificare che le colonne contengano NULL.

```
select * from venue where venuestate is null;

 venueid |        venuename         | venuecity | venuestate | venueseats
---------+--------------------------+-----------+------------+------------
      22 | Quicken Loans Arena      | Cleveland |            |          0
```

```
     101 | Progressive Field        | Cleveland |            |      43345
      72 | Cleveland Browns Stadium | Cleveland |            |      73200
```

Ora, specifica UNLOAD per la tabella VENUE usando l'opzione NULL AS per sostituire i valori nulli con la stringa di caratteri '`fred`'.

```
unload ('select * from venue')
to 's3://mybucket/nulls/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
null as 'fred';
```

Il seguente esempio del file scaricato mostra che i valori nulli sono stati sostituiti con `fred`. Risulta che anche alcuni valori di VENUESEATS erano null e sono stati sostituiti con `fred`. Anche se il tipo di dati di VENUESEATS è intero, UNLOAD converte i valori in testo nei file scaricati, quindi COPY li converte in numeri interi. Se stai scaricando in un file a larghezza fissa, la stringa NULL AS non deve essere più grande della larghezza del campo.

```
248|Charles Playhouse|Boston|MA|0
251|Paris Hotel|Las Vegas|NV|fred
258|Tropicana Hotel|Las Vegas|NV|fred
300|Kennedy Center Opera House|Washington|DC|0
306|Lyric Opera House|Baltimore|MD|0
308|Metropolitan Opera|New York City|NY|0
  5|Gillette Stadium|Foxborough|MA|5
 22|Quicken Loans Arena|Cleveland|fred|0
101|Progressive Field|Cleveland|fred|43345
...
```

Per caricare una tabella dai file scaricati, utilizza un comando COPY con la stessa opzione NULL AS.

> **Note**
>
> Se tenti di caricare i null in una colonna definita come NON NULL, il comando COPY ha esito negativo.

```
create table loadvenuenulls (like venue);

copy loadvenuenulls from 's3://mybucket/nulls/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
```

```
null as 'fred';
```

Per verificare che le colonne contengano null, non solo stringhe vuote, seleziona da LOADVENUENULLS e filtra per null.

```
select * from loadvenuenulls where venuestate is null or venueseats is null;

 venueid |        venuename         | venuecity | venuestate | venueseats
---------+--------------------------+-----------+------------+------------
      72 | Cleveland Browns Stadium | Cleveland |            |      73200
     253 | Mirage Hotel             | Las Vegas | NV         |
     255 | Venetian Hotel           | Las Vegas | NV         |
      22 | Quicken Loans Arena      | Cleveland |            |          0
     101 | Progressive Field        | Cleveland |            |      43345
     251 | Paris Hotel              | Las Vegas | NV         |

...
```

Puoi usare UNLOAD con una tabella che contiene valori null utilizzando il comportamento predefinito NULL AS e quindi specificare COPY per copiare i dati di nuovo in una tabella utilizzando il comportamento predefinito NULL AS; tuttavia, tutti i campi non numerici nella tabella di destinazione contengono stringhe vuote, non null. Per impostazione predefinita, UNLOAD converte i valori null in stringhe vuote (spazio vuoto o lunghezza zero). COPY converte le stringhe vuote in NULL per le colonne numeriche, ma inserisce stringhe vuote in colonne non numeriche. L'esempio seguente mostra come eseguire un comando UNLOAD seguito da COPY utilizzando il comportamento predefinito NULL AS.

```
unload ('select * from venue')
to 's3://mybucket/nulls/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole' allowoverwrite;

truncate loadvenuenulls;
copy loadvenuenulls from 's3://mybucket/nulls/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole';
```

In questo caso, quando filtri per null, solo le righe in cui VENUESEATS contiene valori nulli. Dove VENUESTATE contiene valori nulli nella tabella (VENUE), VENUESTATE nella tabella di destinazione (LOADVENUENULLS) contiene stringhe vuote.

```
select * from loadvenuenulls where venuestate is null or venueseats is null;
```

```
 venueid |        venuename         | venuecity | venuestate | venueseats
---------+--------------------------+-----------+------------+------------
     253 | Mirage Hotel             | Las Vegas | NV         |
     255 | Venetian Hotel           | Las Vegas | NV         |
     251 | Paris Hotel              | Las Vegas | NV         |
...
```

Per caricare le stringhe vuote in colonne non numeriche come NULL, includi le opzioni EMPTYASNULL o BLANKSASNULL. Va bene usarle entrambe.

```
unload ('select * from venue')
to 's3://mybucket/nulls/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole' allowoverwrite;

truncate loadvenuenulls;
copy loadvenuenulls from 's3://mybucket/nulls/'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole' EMPTYASNULL;
```

Per verificare che le colonne contengano NULL e non solo stringhe vuote o spazi, selezionare da LOADVENUENULLS e filtrare per null.

```
select * from loadvenuenulls where venuestate is null or venueseats is null;

 venueid |        venuename         | venuecity | venuestate | venueseats
---------+--------------------------+-----------+------------+------------
      72 | Cleveland Browns Stadium | Cleveland |            |      73200
     253 | Mirage Hotel             | Las Vegas | NV         |
     255 | Venetian Hotel           | Las Vegas | NV         |
      22 | Quicken Loans Arena      | Cleveland |            |          0
     101 | Progressive Field        | Cleveland |            |      43345
     251 | Paris Hotel              | Las Vegas | NV         |
     ...
```

Scarica utilizzando il parametro ALLOWOVERWRITE

Per impostazione predefinita, UNLOAD non sovrascrive i file esistenti nel bucket di destinazione. Ad esempio, se esegui la stessa istruzione UNLOAD due volte senza modificare i file nel bucket di destinazione, il secondo UNLOAD non riesce. Per sovrascrivere i file esistenti, incluso il file manifest, specifica l'opzione ALLOWOVERWRITE.

```
unload ('select * from venue')
```

```
to 's3://mybucket/venue_pipe_'
iam_role 'arn:aws:iam::0123456789012:role/MyRedshiftRole'
manifest allowoverwrite;
```

Scarica la tabella EVENT utilizzando i parametri PARALLEL e MANIFEST

È possibile SCARICARE una tabella in parallelo e generare un file manifest. I file di dati Amazon S3 sono tutti creati allo stesso livello e ai nomi viene aggiunto il suffisso dello schema `0000_part_00`. Il file manifest si trova allo stesso livello di cartella dei file di dati e ha il suffisso del testo `manifest`. Il seguente codice SQL scarica la tabella EVENT e crea file con il nome di base `parallel`

```
unload ('select * from mytickit1.event')
to 's3://my-s3-bucket-name/parallel'
iam_role 'arn:aws:iam::123456789012:role/MyRedshiftRole'
parallel on
manifest;
```

L'elenco dei file Amazon S3 è simile al seguente.

```
 Name                        Last modified                          Size
 parallel0000_part_00 -   August 2, 2023, 14:54:39 (UTC-07:00) 52.1 KB
 parallel0001_part_00 -   August 2, 2023, 14:54:39 (UTC-07:00) 53.4 KB
 parallel0002_part_00 -   August 2, 2023, 14:54:39 (UTC-07:00) 52.1 KB
 parallel0003_part_00 -   August 2, 2023, 14:54:39 (UTC-07:00) 51.1 KB
 parallel0004_part_00 -   August 2, 2023, 14:54:39 (UTC-07:00) 54.6 KB
 parallel0005_part_00 -   August 2, 2023, 14:54:39 (UTC-07:00) 53.4 KB
 parallel0006_part_00 -   August 2, 2023, 14:54:39 (UTC-07:00) 54.1 KB
 parallel0007_part_00 -   August 2, 2023, 14:54:39 (UTC-07:00) 55.9 KB
 parallelmanifest       -   August 2, 2023, 14:54:39 (UTC-07:00) 886.0 B
```

Il contenuto del file `parallelmanifest` è simile al seguente:

```
{
  "entries": [
    {"url":"s3://my-s3-bucket-name/parallel0000_part_00", "meta": { "content_length":
 53316 }},
    {"url":"s3://my-s3-bucket-name/parallel0001_part_00", "meta": { "content_length":
 54704 }},
```

```
    {"url":"s3://my-s3-bucket-name/parallel0002_part_00", "meta": { "content_length":
 53326 }},
    {"url":"s3://my-s3-bucket-name/parallel0003_part_00", "meta": { "content_length":
 52356 }},
    {"url":"s3://my-s3-bucket-name/parallel0004_part_00", "meta": { "content_length":
 55933 }},
    {"url":"s3://my-s3-bucket-name/parallel0005_part_00", "meta": { "content_length":
 54648 }},
    {"url":"s3://my-s3-bucket-name/parallel0006_part_00", "meta": { "content_length":
 55436 }},
    {"url":"s3://my-s3-bucket-name/parallel0007_part_00", "meta": { "content_length":
 57272 }}
  ]
}
```

Scarica la tabella EVENT utilizzando i parametri PARALLEL OFF e MANIFEST

È possibile SCARICARE una tabella in parallelo (PARALLEL OFF) e generare un file manifest. I file di dati Amazon S3 sono tutti creati allo stesso livello e ai nomi viene aggiunto il suffisso dello schema `0000`. Il file manifest si trova allo stesso livello di cartella dei file di dati e ha il suffisso del testo `manifest`.

```
unload ('select * from mytickit1.event')
to 's3://my-s3-bucket-name/serial'
iam_role 'arn:aws:iam::123456789012:role/MyRedshiftRole'
parallel off
manifest;
```

L'elenco dei file Amazon S3 è simile al seguente.

```
 Name                    Last modified                     Size
 serial0000          -   August 2, 2023, 15:54:39 (UTC-07:00) 426.7 KB
 serialmanifest      -   August 2, 2023, 15:54:39 (UTC-07:00) 120.0 B
```

Il contenuto del file `serialmanifest` è simile al seguente:

```
{
  "entries": [
    {"url":"s3://my-s3-bucket-name/serial000", "meta": { "content_length": 436991 }}
```

```
  ]
}
```

Scarica la tabella EVENT utilizzando i parametri PARTITION BY e MANIFEST

È possibile SCARICARE una tabella per partizione e generare un file manifest. In Amazon S3 viene creata una nuova cartella con cartelle di partizione secondarie e i file di dati nelle cartelle secondarie con uno schema di nome simile a `0000_par_00`. Il file manifest si trova allo stesso livello di cartella delle cartelle secondarie con il nome `manifest`.

```
unload ('select * from mytickit1.event')
to 's3://my-s3-bucket-name/partition'
iam_role 'arn:aws:iam::123456789012:role/MyRedshiftRole'
partition by (eventname)
manifest;
```

L'elenco dei file Amazon S3 è simile al seguente.

```
Name                    Type     Last modified                     Size

partition            Folder
```

Nella cartella `partition` ci sono cartelle secondarie con il nome della partizione e il file manifest. Di seguito è riportata la parte inferiore dell'elenco delle cartelle nella cartella `partition`, simile alla seguente.

```
Name                    Type      Last modified                      Size

...
eventname=Zucchero/    Folder
eventname=Zumanity/    Folder
eventname=ZZ Top/      Folder
manifest               -     August 2, 2023, 15:54:39 (UTC-07:00) 467.6 KB
```

Nella cartella `eventname=Zucchero/` ci sono i file di dati simili ai seguenti.

```
 Name               Last modified                          Size
 0000_part_00 -   August 2, 2023, 15:59:19 (UTC-07:00) 70.0 B
 0001_part_00 -   August 2, 2023, 15:59:16 (UTC-07:00) 106.0 B
 0002_part_00 -   August 2, 2023, 15:59:15 (UTC-07:00) 70.0 B
 0004_part_00 -   August 2, 2023, 15:59:17 (UTC-07:00) 141.0 B
 0006_part_00 -   August 2, 2023, 15:59:16 (UTC-07:00) 35.0 B
 0007_part_00 -   August 2, 2023, 15:59:19 (UTC-07:00) 108.0 B
```

L'aspetto del contenuto del file `manifest` è simile al seguente:

```
{
  "entries": [
    ...
    {"url":"s3://my-s3-bucket-name/partition/eventname=Zucchero/0007_part_00", "meta":
 { "content_length": 108 }},
    {"url":"s3://my-s3-bucket-name/partition/eventname=Zumanity/0007_part_00", "meta":
 { "content_length": 72 }}
  ]
}
```

Scarica la tabella EVENT utilizzando i parametri MAXFILESIZE, ROWGROUPSIZE e MANIFEST

È possibile SCARICARE una tabella in parallelo e generare un file manifest. I file di dati Amazon S3 sono tutti creati allo stesso livello e ai nomi viene aggiunto il suffisso dello schema `0000_part_00`. I file di dati Parquet generati sono limitati a 256 MB e la dimensione del gruppo di righe è di 128 MB. Il file manifest si trova allo stesso livello di cartella dei file di dati e ha il suffisso `manifest`.

```
unload ('select * from mytickit1.event')
to 's3://my-s3-bucket-name/eventsize'
iam_role 'arn:aws:iam::123456789012:role/MyRedshiftRole'
maxfilesize 256 MB
rowgroupsize 128 MB
parallel on
parquet
manifest;
```

L'elenco dei file Amazon S3 è simile al seguente.

```
 Name                            Type      Last modified                        Size
 eventsize0000_part_00.parquet parquet August 2, 2023, 17:35:21 (UTC-07:00) 24.5 KB
```

```
 eventsize0001_part_00.parquet parquet August 2, 2023, 17:35:21 (UTC-07:00) 24.8 KB
 eventsize0002_part_00.parquet parquet August 2, 2023, 17:35:21 (UTC-07:00) 24.4 KB
 eventsize0003_part_00.parquet parquet August 2, 2023, 17:35:21 (UTC-07:00) 24.0 KB
 eventsize0004_part_00.parquet parquet August 2, 2023, 17:35:21 (UTC-07:00) 25.3 KB
 eventsize0005_part_00.parquet parquet August 2, 2023, 17:35:21 (UTC-07:00) 24.8 KB
 eventsize0006_part_00.parquet parquet August 2, 2023, 17:35:21 (UTC-07:00) 25.0 KB
 eventsize0007_part_00.parquet parquet August 2, 2023, 17:35:21 (UTC-07:00) 25.6 KB
 eventsizemanifest                 -       August 2, 2023, 17:35:21 (UTC-07:00) 958.0 B
```

Il contenuto del file `eventsizemanifest` è simile al seguente:

```
{
  "entries": [
    {"url":"s3://my-s3-bucket-name/eventsize0000_part_00.parquet", "meta":
 { "content_length": 25130 }},
    {"url":"s3://my-s3-bucket-name/eventsize0001_part_00.parquet", "meta":
 { "content_length": 25428 }},
    {"url":"s3://my-s3-bucket-name/eventsize0002_part_00.parquet", "meta":
 { "content_length": 25025 }},
    {"url":"s3://my-s3-bucket-name/eventsize0003_part_00.parquet", "meta":
 { "content_length": 24554 }},
    {"url":"s3://my-s3-bucket-name/eventsize0004_part_00.parquet", "meta":
 { "content_length": 25918 }},
    {"url":"s3://my-s3-bucket-name/eventsize0005_part_00.parquet", "meta":
 { "content_length": 25362 }},
    {"url":"s3://my-s3-bucket-name/eventsize0006_part_00.parquet", "meta":
 { "content_length": 25647 }},
    {"url":"s3://my-s3-bucket-name/eventsize0007_part_00.parquet", "meta":
 { "content_length": 26256 }}
  ]
}
```

UPDATE

Argomenti

- Sintassi
- Parametri
- Note per l'utilizzo
- Esempi di istruzioni UPDATE

Aggiorna i valori in una o più colonne della tabella quando una condizione è soddisfatta.

> **Note**
>
> Le dimensioni massime per una istruzione SQL è di 16 MB.

Sintassi

```
[ WITH [RECURSIVE] common_table_expression [, common_table_expression , ...] ]
            UPDATE table_name [ [ AS ] alias ] SET column = { expression | DEFAULT }
 [,...]

[ FROM fromlist ]
[ WHERE condition ]
```

Parametri

Clausola WITH

Clausola facoltativa che specifica una o più common-table-expressions. Per informazioni, consultare Clausola WITH.

table_name

Tabella temporanea o persistente. Solo il proprietario della tabella o un utente con il privilegio UPDATE sulla tabella può aggiornare le righe. Se utilizzi la clausola FROM o selezioni dalle tabelle in un'espressione o una condizione, devi avere il privilegio SELECT su tali tabelle. Non puoi assegnare alla tabella un alias, ma puoi specificare un alias nella clausola FROM.

> **Note**
>
> Le tabelle esterne di Amazon Redshift Spectrum sono di sola lettura. Non puoi eseguire UPDATE per una tabella esterna.

alias

Nome alternativo temporaneo per una tabella di destinazione. Gli alias sono facoltativi. La parola chiave AS è sempre facoltativa.

SET column =

Una o più colonne che vuoi modificare. Le colonne che non sono elencate mantengono i rispettivi valori correnti. Non includere il nome della tabella nella specifica di una colonna di destinazione. Ad esempio, `UPDATE tab SET tab.col = 1` non è valido.

espressione

Espressione che definisce il nuovo valore per la colonna specificata.

DEFAULT

Aggiorna la colonna con il valore predefinito assegnato alla colonna nell'istruzione CREATE TABLE.

FROM tablelist

Puoi aggiornare una tabella facendo riferimento alle informazioni presenti in altre tabelle. Elenca queste altre tabelle nella clausola FROM o utilizza una sottoquery come parte della condizione WHERE. Le tabelle elencate nella clausola FROM possono avere alias. Se è necessario includere la tabella di destinazione dell'istruzione UPDATE nell'elenco, utilizza un alias.

WHERE condition

Clausola facoltativa che limita gli aggiornamenti alle righe che corrispondono a una condizione. Quando la condizione restituisce `true`, le colonne SET specificate vengono aggiornate. La condizione può essere un semplice predicato su una colonna o una condizione basata sul risultato di una sottoquery.

Puoi nominare qualsiasi tabella nella sottoquery, inclusa la tabella di destinazione per UPDATE.

Note per l'utilizzo

Dopo aver aggiornato un numero elevato di righe in una tabella:

- Applica l'operazione di vacuum alla tabella per recuperare spazio di memorizzazione e riordinare le righe.
- Analizza la tabella per aggiornare le statistiche del pianificatore di query.

Gli outer join sinistra, destra e completi non sono supportati nella clausola FROM di un'istruzione UPDATE e restituiscono il seguente errore:

```
ERROR: Target table must be part of an equijoin predicate
```

Se è necessario specificare un outer join, utilizza una sottoquery nella clausola WHERE dell'istruzione UPDATE.

Se l'istruzione UPDATE richiede un self join alla tabella di destinazione, è necessario specificare la condizione di join e i criteri della clausola WHERE che qualificano le righe per l'operazione di aggiornamento. In generale, quando la tabella di destinazione viene unita a se stessa o a un'altra tabella, una best practice è utilizzare una sottoquery che separa chiaramente le condizioni di join dai criteri che qualificano le righe per gli aggiornamenti.

Le query UPDATE con più corrispondenze per riga generano un errore quando il parametro di configurazione `error_on_nondeterministic_update` è impostato su true. Per ulteriori informazioni, consulta error_on_nondeterministic_update.

Puoi aggiornare una colonna GENERATED BY DEFAULT AS IDENTITY. Le colonne definite come GENERATED BY DEFAULT AS IDENTITY possono essere aggiornate con i valori forniti. Per ulteriori informazioni, consulta GENERATED BY DEFAULT AS IDENTITY.

Esempi di istruzioni UPDATE

Per ulteriori informazioni sulle tabelle utilizzate negli esempi seguenti, consultare Database di esempio.

La tabella CATEGORY nel database TICKIT contiene le seguenti righe:

```
+-------+----------+-----------+--------------------------------------------+
| catid | catgroup |  catname  |                  catdesc                   |
+-------+----------+-----------+--------------------------------------------+
| 5     | Sports   | MLS       | Major League Soccer                        |
| 11    | Concerts | Classical | All symphony, concerto, and choir concerts |
| 1     | Sports   | MLB       | Major League Baseball                      |
| 6     | Shows    | Musicals  | Musical theatre                            |
| 3     | Sports   | NFL       | National Football League                   |
| 8     | Shows    | Opera     | All opera and light opera                  |
| 2     | Sports   | NHL       | National Hockey League                     |
| 9     | Concerts | Pop       | All rock and pop music concerts            |
| 4     | Sports   | NBA       | National Basketball Association            |
| 7     | Shows    | Plays     | All non-musical theatre                    |
| 10    | Concerts | Jazz      | All jazz singers and bands                 |
+-------+----------+-----------+--------------------------------------------+
```

Aggiornamento di una tabella in base a un intervallo di valori

Aggiorna la colonna CATGROUP in base a un intervallo di valori nella colonna CATID.

```
UPDATE category
SET catgroup='Theatre'
WHERE catid BETWEEN 6 AND 8;

SELECT * FROM category
WHERE catid BETWEEN 6 AND 8;

+-------+----------+----------+--------------------------+
| catid | catgroup | catname  |          catdesc         |
+-------+----------+----------+--------------------------+
| 6     | Theatre  | Musicals | Musical theatre          |
| 7     | Theatre  | Plays    | All non-musical theatre  |
| 8     | Theatre  | Opera    | All opera and light opera |
+-------+----------+----------+--------------------------+
```

Aggiornamento di una tabella in base a un valore corrente

Aggiorna le colonne CATNAME e CATDESC in base al valore corrente di CATGROUP:

```
UPDATE category
SET catdesc=default, catname='Shows'
WHERE catgroup='Theatre';

SELECT * FROM category
WHERE catname='Shows';

+-------+----------+---------+---------+
| catid | catgroup | catname | catdesc |
+-------+----------+---------+---------+
| 6     | Theatre  | Shows   | NULL    |
| 7     | Theatre  | Shows   | NULL    |
| 8     | Theatre  | Shows   | NULL    |
+-------+----------+---------+---------+)
```

In questo caso, la colonna CATDESC è stata impostata su null perché non è stato definito alcun valore predefinito quando è stata creata la tabella.

Esegui i seguenti comandi per reimpostare i dati della tabella CATEGORY sui valori originali:

```
TRUNCATE category;

COPY category
FROM 's3://redshift-downloads/tickit/category_pipe.txt'
DELIMITER '|'
IGNOREHEADER 1
REGION 'us-east-1'
IAM_ROLE default;
```

Aggiornamento di una tabella in base al risultato di una sottoquery della clausola WHERE

Aggiorna la tabella CATEGORY in base al risultato di una sottoquery nella clausola WHERE:

```
UPDATE category
SET catdesc='Broadway Musical'
WHERE category.catid IN
(SELECT category.catid FROM category
JOIN event ON category.catid = event.catid
JOIN venue ON venue.venueid = event.venueid
JOIN sales ON sales.eventid = event.eventid
WHERE venuecity='New York City' AND catname='Musicals');
```

Visualizza la tabella aggiornata:

```
SELECT * FROM category ORDER BY catid;

+-------+----------+-----------+--------------------------------------------+
| catid | catgroup |  catname  |                  catdesc                   |
+-------+----------+-----------+--------------------------------------------+
| 2     | Sports   | NHL       | National Hockey League                     |
| 3     | Sports   | NFL       | National Football League                   |
| 4     | Sports   | NBA       | National Basketball Association            |
| 5     | Sports   | MLS       | Major League Soccer                        |
| 6     | Shows    | Musicals  | Broadway Musical                           |
| 7     | Shows    | Plays     | All non-musical theatre                    |
| 8     | Shows    | Opera     | All opera and light opera                  |
| 9     | Concerts | Pop       | All rock and pop music concerts            |
| 10    | Concerts | Jazz      | All jazz singers and bands                 |
| 11    | Concerts | Classical | All symphony, concerto, and choir concerts |
+-------+----------+-----------+--------------------------------------------+
```

Aggiornamento di una tabella in base al risultato di una sottoquery della clausola WITH

Per aggiornare la tabella CATEGORY in base al risultato di una sottoquery utilizzando la clausola WITH, utilizza l'esempio seguente.

```
WITH u1 as (SELECT catid FROM event ORDER BY catid DESC LIMIT 1)
UPDATE category SET catid='200' FROM u1 WHERE u1.catid=category.catid;

SELECT * FROM category ORDER BY catid DESC LIMIT 1;

+-------+----------+---------+---------------------------------+
| catid | catgroup | catname |             catdesc             |
+-------+----------+---------+---------------------------------+
| 200   | Concerts | Pop     | All rock and pop music concerts |
+-------+----------+---------+---------------------------------+
```

Aggiornamento di una tabella in base al risultato di una condizione di join

Aggiorna le 11 righe originali nella tabella CATEGORY in base alle righe CATID corrispondenti nella tabella EVENT:

```
UPDATE category SET catid=100
FROM event
WHERE event.catid=category.catid;

SELECT * FROM category ORDER BY catid;

+-------+----------+-----------+---------------------------------------------+
| catid | catgroup |  catname  |                  catdesc                    |
+-------+----------+-----------+---------------------------------------------+
| 2     | Sports   | NHL       | National Hockey League                      |
| 3     | Sports   | NFL       | National Football League                    |
| 4     | Sports   | NBA       | National Basketball Association             |
| 5     | Sports   | MLS       | Major League Soccer                         |
| 10    | Concerts | Jazz      | All jazz singers and bands                  |
| 11    | Concerts | Classical | All symphony, concerto, and choir concerts  |
| 100   | Concerts | Pop       | All rock and pop music concerts             |
| 100   | Shows    | Plays     | All non-musical theatre                     |
| 100   | Shows    | Opera     | All opera and light opera                   |
| 100   | Shows    | Musicals  | Broadway Musical                            |
+-------+----------+-----------+---------------------------------------------+
```

Tieni presente che la tabella EVENT è elencata nella clausola FROM e la condizione di join alla tabella di destinazione è definita nella clausola WHERE. Solo quattro righe sono qualificate per

l'aggiornamento. Queste quattro righe sono le righe i cui i valori CATID erano originariamente 6, 7, 8 e 9; solo queste quattro categorie sono rappresentate nella tabella EVENT:

```
SELECT DISTINCT catid FROM event;

+-------+
| catid |
+-------+
| 6     |
| 7     |
| 8     |
| 9     |
+-------+
```

Aggiorna le 11 righe originali nella tabella CATEGORY estendendo l'esempio precedente e aggiungendo un'altra condizione alla clausola WHERE. A causa della restrizione sulla colonna CATGROUP, solo una riga si qualifica per l'aggiornamento (sebbene quattro righe siano idonee per il join).

```
UPDATE category SET catid=100
FROM event
WHERE event.catid=category.catid
AND catgroup='Concerts';

SELECT * FROM category WHERE catid=100;

+-------+----------+---------+---------------------------------+
| catid | catgroup | catname |             catdesc             |
+-------+----------+---------+---------------------------------+
| 100   | Concerts | Pop     | All rock and pop music concerts |
+-------+----------+---------+---------------------------------+
```

Un modo alternativo per scrivere questo esempio è il seguente:

```
UPDATE category SET catid=100
FROM event JOIN category cat ON event.catid=cat.catid
WHERE cat.catgroup='Concerts';
```

Il vantaggio di questo approccio è che i criteri di join sono chiaramente separati da qualsiasi altro criterio che qualifica le righe per l'aggiornamento. Nota l'uso dell'alias CAT per la tabella CATEGORY nella clausola FROM.

Aggiornamenti con outer join nella clausola FROM

L'esempio precedente mostrava un inner join specificato nella clausola FROM di un'istruzione UPDATE. L'esempio seguente restituisce un errore perché la clausola FROM non supporta gli outer join alla tabella di destinazione:

```
UPDATE category SET catid=100
FROM event LEFT JOIN category cat ON event.catid=cat.catid
WHERE cat.catgroup='Concerts';
ERROR:  Target table must be part of an equijoin predicate
```

Se è necessario specificare un outer join per l'istruzione UPDATE, puoi spostare la sintassi dell'outer join in una sottoquery.

```
UPDATE category SET catid=100
FROM
(SELECT event.catid FROM event LEFT JOIN category cat ON event.catid=cat.catid)
 eventcat
WHERE category.catid=eventcat.catid
AND catgroup='Concerts';
```

Aggiornamenti con colonne di un'altra tabella nella clausola SET

Utilizza l'esempio seguente per aggiornare la tabella listing nel database di esempio TICKIT con i valori della tabella sales.

```
SELECT listid, numtickets FROM listing WHERE sellerid = 1 ORDER BY 1 ASC LIMIT 5;

+--------+------------+
| listid | numtickets |
+--------+------------+
| 100423 | 4          |
| 108334 | 24         |
| 117150 | 4          |
| 135915 | 20         |
| 205927 | 6          |
+--------+------------+

UPDATE listing
SET numtickets = sales.sellerid
FROM sales
```

```
WHERE sales.sellerid = 1 AND listing.sellerid = sales.sellerid;

SELECT listid, numtickets FROM listing WHERE sellerid = 1 ORDER BY 1 ASC LIMIT 5;

+--------+------------+
| listid | numtickets |
+--------+------------+
| 100423 | 1          |
| 108334 | 1          |
| 117150 | 1          |
| 135915 | 1          |
| 205927 | 1          |
+--------+------------+
```

VACUUM

Riordina le righe e recupera spazio di memorizzazione di una tabella specificata o di tutte le tabelle del database corrente.

> **Note**
>
> Solo gli utenti con le autorizzazioni necessarie possono efficacemente eseguire l'operazione vacuum su una tabella. Se VACUUM viene eseguito senza le necessarie autorizzazioni di tabella, l'operazione viene completata ma non ha alcun effetto. Per l'elenco delle autorizzazioni di tabella valide per eseguire efficacemente VACUUM, consulta la seguente sezione Privilegi richiesti.

Amazon Redshift riordina automaticamente i dati ed esegue VACUUM DELETE in background. Ciò riduce la necessità di eseguire il comando VACUUM. Per ulteriori informazioni, consulta Vacuum delle tabelle.

Per impostazione predefinita, il comando VACUUM ignora la fase di ordinamento per ogni tabella dove più del 95% delle righe della tabella sono già ordinate. Ignorare la fase di ordinamento può migliorare significativamente le prestazioni di VACUUM. Per modificare la soglia predefinita di ordinamento o eliminazione per una singola tabella, includi il nome della tabella e il parametro TO threshold PERCENT quando esegui il comando VACUUM.

Gli utenti possono accedere alle tabelle mentre ne viene eseguito il vacuum. È possibile eseguire query e scrivere operazioni mentre viene eseguito il vacuum di una tabella, ma quando i comandi

DML (Data Manipulation Language) e il vacuum vengono eseguiti contemporaneamente, potrebbero richiedere più tempo. Se si eseguono le istruzioni UPDATE e DELETE durante un vacuum, le prestazioni del sistema potrebbero essere ridotte. VACUUM DELETE blocca temporaneamente le operazioni di aggiornamento ed eliminazione.

Amazon Redshift esegue automaticamente un vacuum DELETE ONLY in background. L'operazione di vacuum automatica viene sospesa quando gli utenti eseguono operazioni DDL (Data Definition Language), ad esempio ALTER TABLE.

> **Note**
>
> La sintassi e il comportamento del comando VACUUM in Amazon Redshift sono sostanzialmente diversi dall'operazione VACUUM di PostgreSQL. Ad esempio, l'operazione VACUUM di default in Amazon Redshift è VACUUM FULL, che recupera spazio su disco e riordina tutte le righe. Al contrario, l'operazione VACUUM predefinita in PostgreSQL recupera solo lo spazio e lo rende disponibile per il riutilizzo.

Per ulteriori informazioni, consulta Vacuum delle tabelle.

Privilegi richiesti

Di seguito sono elencati i privilegi necessari per VACUUM:

- Superuser
- Utenti con il privilegio VACUUM
- Proprietario della tabella
- Proprietario del database con cui è condivisa la tabella

Sintassi

```
VACUUM [ FULL | SORT ONLY | DELETE ONLY | REINDEX | RECLUSTER ]
[ [ table_name ] [ TO threshold PERCENT ] [ BOOST ] ]
```

Parametri

FULL

Ordina la tabella specificata (o tutte le tabelle nel database corrente) e recupera lo spazio su disco occupato dalle righe che sono state contrassegnate per l'eliminazione dalle precedenti operazioni UPDATE e DELETE. VACUUM FULL è il valore predefinito.

Un'operazione vacuum full non esegue una reindicizzazione delle tabelle interlacciate. Per reindicizzare le tabelle interlacciate seguite da un vacuum full, utilizza l'opzione VACUUM REINDEX.

Per impostazione predefinita, il comando VACUUM FULL ignora la fase di ordinamento per ogni tabella con almeno il 95% delle righe ordinate. Se VACUUM è in grado di ignorare la fase di ordinamento, esegue il comando DELETE ONLY e recupera lo spazio nella fase di eliminazione in modo tale che almeno il 95 percento delle restanti righe non sia contrassegnato per l'eliminazione.

Se la soglia di ordinamento non viene soddisfatta (ad esempio, se il 90 percento delle righe viene ordinato) e VACUUM esegue un ordinamento completo, viene eseguita anche un'operazione di eliminazione completa, recuperando spazio dal 100% delle righe eliminate.

Puoi modificare la soglia di vacuum predefinita solo per una singola tabella. Per modificare la soglia predefinita di vacuum per una singola tabella, includi il nome della tabella e il parametro TO threshold PERCENT.

SORT ONLY

Ordina la tabella specificata (o tutte le tabelle nel database corrente) senza recuperare spazio liberato dalle righe eliminate. Questa opzione è utile quando il recupero di spazio su disco non è importante, ma è importante riordinare le nuove righe. Un vacuum SORT ONLY riduce il tempo trascorso per le operazioni di vacuum quando la regione non ordinata non contiene un numero elevato di righe eliminate e non copre l'intera regione ordinata. Le applicazioni che non hanno vincoli di spazio su disco ma dipendono dalle ottimizzazioni delle query associate al mantenimento ordinato delle righe della tabella possono trarre vantaggio da questo tipo di vacuum.

Per impostazione predefinita, il comando VACUUM SORT ONLY ignora ogni tabella con almeno il 95% delle righe ordinate. Per modificare la soglia predefinita di ordinamento per una singola tabella, includi il nome della tabella e il parametro TO threshold PERCENT quando esegui il comando VACUUM.

DELETE ONLY

Amazon Redshift esegue automaticamente un vacuum DELETE ONLY in background, per cui non sarà quasi mai necessario eseguire un vacuum DELETE ONLY.

Un VACUUM DELETE recupera lo spazio su disco occupato dalle righe che sono state contrassegnate per l'eliminazione dalle precedenti operazioni UPDATE e DELETE e compatta la tabella per liberare lo spazio utilizzato. Un'operazione di vacuum DELETE ONLY non ordina i dati della tabella.

Questa opzione riduce il tempo necessario per le operazioni di vacuum quando il recupero di spazio su disco è importante, ma non è importante riordinare le nuove righe. Questa opzione può anche essere utile quando le prestazioni della query sono già ottimali e il riordinamento delle righe per ottimizzare le prestazioni della query non è un requisito.

Per impostazione predefinita, VACUUM DELETE ONLY consente di recuperare spazio in modo tale che almeno il 95% delle restanti righe non viene contrassegnato per l'eliminazione. Per modificare la soglia predefinita di eliminazione per una singola tabella, includi il nome della tabella e il parametro TO threshold PERCENT quando esegui il comando VACUUM.

Alcune operazioni, come `ALTER TABLE APPEND`, possono causare la frammentazione delle tabelle. Quando usi la clausola `DELETE ONLY`, l'operazione di vacuum recupera lo spazio dalle tabelle frammentate. Lo stesso valore di soglia del 95 percento si applica all'operazione di deframmentazione.

REINDEX tablename

Analizza la distribuzione dei valori nelle colonne interlacciate della chiave di ordinamento, quindi esegue un'operazione di VACUUM completa. Se si utilizza REINDEX, è necessario un nome di tabella.

VACUUM REINDEX può impiegare molto più tempo rispetto a VACUUM FULL perché esegue un ulteriore passaggio per analizzare le chiavi di ordinamento interlacciato. L'operazione di ordinamento e unione può richiedere più tempo per le tabelle interlacciate perché l'ordinamento interlacciato potrebbe dover riordinare più righe rispetto a un ordinamento composto.

Se un'operazione di VACUUM REINDEX termina prima del completamento, il successivo VACUUM riprende l'operazione di reindicizzazione prima di eseguire l'operazione vacuum full.

VACUUM REINDEX non è supportato con TO threshold PERCENT.

table_name

Nome della tabella da sottoporre al vacuum. Se non si specifica un nome tabella, l'operazione di vacuum si applica a tutte le tabelle nel database corrente. Puoi specificare qualsiasi tabella permanente o temporanea creata dall'utente. Il comando non è significativo per altri oggetti, come le viste e le tabelle di sistema.

Se includi il parametro TO threshold PERCENT, è richiesto un nome tabella.

RECLUSTER nometabella

Ordina le parti della tabella che non sono ordinate. Le parti della tabella che sono già ordinate in base all'ordinamento automatico della tabella rimangono intatte. Questo comando non unisce i dati appena ordinati con la regione ordinata. Inoltre, non recupera tutto lo spazio contrassegnato per l'eliminazione. Al termine di questo comando, la tabella potrebbe non apparire completamente ordinata, come indicato dal campo `unsorted` in SVV_TABLE_INFO.

Si consiglia di utilizzare VACUUM RECLUSTER per tabelle di grandi dimensioni con importazione frequente e query che accedono solo ai dati più recenti.

VACUUM RECLUSTER non è supportato con TO threshold PERCENT. Se si utilizza RECLUSTER, è necessario un nome di tabella.

VACUY RECLUSTER non è supportato su tabelle con chiavi di ordinamento interlacciato e tabelle con stile di distribuzione ALL.

table_name

Nome della tabella da sottoporre al vacuum. Puoi specificare qualsiasi tabella permanente o temporanea creata dall'utente. Il comando non è significativo per altri oggetti, come le viste e le tabelle di sistema.

TO threshold PERCENT

Clausola che specifica la soglia oltre la quale VACUUM ignora la fase di ordinamento e la soglia di destinazione per recuperare lo spazio nella fase di eliminazione. La soglia di ordinamento è la percentuale delle righe totali che sono già ordinate per la tabella specificata prima del vacuuming. La soglia di eliminazione è la percentuale minima delle righe totali non contrassegnate per l'eliminazione dopo il vacuum.

Dal momento che VACUUM riordina le righe solo quando la percentuale delle righe ordinate in una tabella è inferiore alla soglia di ordinamento, Amazon Redshift può spesso ridurre significativamente i tempi di esecuzione di VACUUM. Analogamente, quando VACUUM non è

vincolato a recuperare spazio dal 100% delle righe contrassegnate per l'eliminazione, è spesso in grado di ignorare i blocchi di riscrittura che contengono solo alcune righe eliminate.

Ad esempio, se specifichi 75 per threshold, VACUUM ignora la fase di ordinamento se almeno il 75% delle righe della tabella sono già ordinate. Per la fase di eliminazione, VACUUMS imposta una destinazione per il recupero dello spazio su disco in modo tale che almeno il 75% delle righe della tabella non sia contrassegnato per l'eliminazione dopo il vacuum. Il valore di threshold deve essere un numero intero compreso tra 0 e 100. Il valore predefinito è 95. Se specifichi il valore 100, VACUUM ordina sempre la tabella a meno che non sia già completamente ordinata e recupera lo spazio da tutte le righe contrassegnate per l'eliminazione. Se specifichi il valore 0, VACUUM non ordina mai la tabella e non recupera lo spazio.

Se includi il parametro TO threshold PERCENT, devi specificare anche un nome tabella. Se non specifichi un nome di tabella, VACUUM non riesce.

Non è possibile utilizzare il parametro TO threshold PERCENT con REINDEX.

BOOST

Esegue il comando VACUUM con risorse aggiuntive, come memoria e spazio su disco, che sono disponibili. Con l'opzione BOOST, VACUUM funziona in una finestra e blocca le eliminazioni e gli aggiornamenti simultanei per la durata dell'operazione VACUUM. L'esecuzione con l'opzione BOOST richiede risorse di sistema, che potrebbero influire sulle prestazioni della query. Eseguire VACUUM BOOST quando il carico sul sistema è leggero, ad esempio durante le operazioni di manutenzione.

Durante l'utilizzo dell'opzione BOOST tenere presente quanto segue:

- Specificando l'opzione BOOST, è obbligatorio specificare il valore table-name.
- L'opzione BOOST non è supportata nell'operazione REINDEX.
- L'opzione BOOST è ignorata nell'operazione DELETE ONLY.

Note per l'utilizzo

Per la maggior parte delle applicazioni Amazon Redshift, si consiglia l'utilizzo di vacuum completo. Per ulteriori informazioni, consulta Vacuum delle tabelle.

Prima di eseguire un'operazione di vacuum, tieni presente il seguente comportamento:

- Non è possibile eseguire VACUUM all'interno di un blocco di transazioni (BEGIN ... END). Per ulteriori informazioni sulle transazioni, consultare Isolamento serializzabile.

- Puoi eseguire un solo comando VACUUM su un cluster in qualsiasi momento. Se si prova a eseguire più operazioni di vacuum contemporaneamente, Amazon Redshift restituisce un errore.
- Le dimensioni della tabella potrebbero crescere quando viene sottoposta al vacuum. Questo comportamento è previsto quando non ci sono righe eliminate da recuperare o il nuovo ordinamento della tabella produce un rapporto inferiore di compressione dei dati.
- Durante le operazioni di vacuum, è previsto un certo degrado delle prestazioni delle query. Le normali prestazioni riprendono non appena l'operazione di vacuum è completa.
- Le operazioni di scrittura simultanee procedono durante le operazioni di vacuum, ma non è consigliabile eseguire operazioni di scrittura durante il vacuum. È più efficiente completare le operazioni di scrittura prima di eseguire il vacuum. Inoltre, tutti i dati scritti dopo l'avvio di un'operazione di vacuum non possono essere resi vacuum da tale operazione. In questo caso, è necessaria una seconda operazione di vacuum.
- Un'operazione di vacuum potrebbe non avviarsi se un'operazione di carico o inserimento è già in corso. Le operazioni di vacuum richiedono temporaneamente l'accesso esclusivo alle tabelle per essere avviate. Questo accesso esclusivo è richiesto per breve tempo, quindi le operazioni di vacuum non bloccano i carichi e gli inserimenti simultanei per un periodo di tempo significativo.
- Le operazioni di vacuum vengono saltate in assenza di lavoro da eseguire per una determinata tabella. Tuttavia, si verifica un sovraccarico associato alla scoperta della possibilità di saltare l'operazione. Se sai che una tabella è una versione originale o non soddisfa la soglia di vacuum, non sottoporla a un'operazione di vacuum.
- Un'operazione di vacuum DELETE ONLY su una piccola tabella potrebbe non ridurre il numero di blocchi utilizzati per memorizzare i dati, specialmente quando la tabella ha un numero elevato di colonne o il cluster utilizza un numero elevato di sezioni per nodo. Queste operazioni di vacuum aggiungono un blocco per colonna per sezione per tenere conto degli inserimenti simultanei nella tabella e c'è il rischio che questo sovraccarico superi la riduzione del numero di blocchi dallo spazio su disco recuperato. Ad esempio, se una tabella con 10 colonne su un cluster con 8 nodi occupa 1000 blocchi prima di un vacuum, il vacuum non riduce il numero di blocchi effettivo a meno che non vengano recuperati più di 80 blocchi di spazio su disco a causa delle righe eliminate. Ogni blocco di dati utilizza 1 MB.

Operazioni di vacuum automatiche in pausa in presenza di una delle condizioni seguenti:

- Un utente esegue un'operazione DDL (Data Definition Language), ad esempio ALTER TABLE, che richiede un blocco esclusivo sulla tabella su cui è attualmente in funzione il vacuum automatico.

- Un utente attiva il VACUUM su una qualsiasi tabella nel cluster (è possibile eseguire un solo VACUUM alla volta).
- Un periodo di caricamento del cluster elevato.

Esempi

Recupera lo spazio del database e riordina le righe in tutte le tabelle in base alla soglia predefinita di vacuum del 95 percento.

```
vacuum;
```

Recupera lo spazio e riordina le righe nella tabella SALES in base alla soglia predefinita del 95 percento.

```
vacuum sales;
```

Recupera sempre lo spazio e riordina le righe nella tabella SALES.

```
vacuum sales to 100 percent;
```

Riordina le righe della tabella SALES solo se risultano già ordinate meno del 75% delle righe.

```
 vacuum sort only sales to 75 percent;
```

Recupera spazio nella tabella SALES in modo tale che almeno il 75 delle restanti righe non sia contrassegnato per l'eliminazione dopo il vacuum.

```
vacuum delete only sales to 75 percent;
```

Reindicizza e quindi sottoponi al vacuum la tabella LISTING.

```
vacuum reindex listing;
```

Il seguente comando restituisce un errore.

```
vacuum reindex listing to 75 percent;
```

Esegui il recluster e quindi sottoponi a vacuum la tabella LISTING.

```
vacuum recluster listing;
```

Esegui il recluster e quindi sottoponi a vacuum la tabella LISTING con l'opzione BOOST.

```
vacuum recluster listing boost;
```

Informazioni di riferimento sulle funzioni SQL

Argomenti

Amazon Redshift supporta un certo numero di funzioni che sono estensioni allo standard SQL, così come funzioni di aggregazione standard, funzioni scalari e funzioni finestra.

> **Note**
>
> Amazon Redshift è basato su PostgreSQL. Amazon Redshift e PostgreSQL si differenziano per un certo numero di aspetti molto importanti, di cui bisogna tener conto mentre si progettano e sviluppano le applicazioni di data warehouse. Per informazioni sulle differenze tra Amazon Redshift SQL e PostgreSQL, consultare Amazon Redshift e PostgreSQL.

Nodo principale: solo funzioni

Alcune query di Amazon Redshift vengono distribuite ed eseguite sui nodi di calcolo; altre vengono eseguite esclusivamente sul nodo principale.

Il nodo principale distribuisce SQL ai nodi di calcolo quando una query fa riferimento a tabelle create dall'utente o a tabelle di sistema (tabelle con prefisso STL o STV e visualizzazioni di sistema con un prefisso SVL o SVV). Una query che fa riferimento solo a tabelle di catalogo (tabelle con un prefisso PG, ad esempio PG_TABLE_DEF) o che non fa riferimento ad alcuna tabella viene eseguita esclusivamente sul nodo principale.

Alcune funzioni SQL di Amazon Redshift sono supportate solo sul nodo principale e non sui nodi di calcolo. Una query che utilizza una funzione del nodo principale deve essere eseguita esclusivamente sul nodo principale e non sui nodi di calcolo, altrimenti verrà restituito un errore.

La documentazione per ciascuna funzione solo sul nodo principale comprende una nota che indica che la funzione restituirà un errore se fa riferimento a tabelle definite dall'utente o a tabelle di sistema di Amazon Redshift.

Per ulteriori informazioni, consultare Funzioni SQL supportate sul nodo principale.

Le seguenti funzioni SQL sono funzioni solo sul nodo principale e non vengono supportate sui nodi di calcolo:

Funzioni di informazioni di sistema

- CURRENT_SCHEMA
- CURRENT_SCHEMAS
- HAS_DATABASE_PRIVILEGE

- PRIVILEGIO DELLO SCHEMA HAS
- HAS_TABLE_PRIVILEGE

Funzioni stringa

- SUBSTR

Funzioni matematiche

- FATTORIALE ()

Le seguenti funzioni solo sul nodo principale sono obsolete e non sono più supportate:

Funzioni di data

- AGE
- CURRENT_TIME
- CURRENT_TIMESTAMP
- LOCALTIME
- ISFINITE
- NOW

Funzioni stringa

- GETBIT
- GET_BYTE
- SET_BIT
- SET_BYTE
- TO_ASCII

Nodo di calcolo: solo funzioni

Alcune query Amazon Redshift devono essere eseguite solo sui nodi di calcolo. Se una query fa riferimento a una tabella creata dall'utente, l'esecuzione di SQL avviene nei nodi di calcolo.

Una query che fa riferimento solo a tabelle di catalogo (tabelle con un prefisso PG, ad esempio PG_TABLE_DEF) o che non fa riferimento ad alcuna tabella viene eseguita esclusivamente sul nodo principale.

Se una query che usa una funzione di un nodo di calcolo non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift, viene restituito il seguente errore.

```
[Amazon](500310) Invalid operation: One or more of the used functions must be applied
 on at least one user created table.
```

La documentazione per ciascuna funzione solo sul nodo di calcolo comprende una nota che indica che la funzione restituirà un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift.

Di seguito sono elencate le funzioni SQL che sono solo funzioni sul nodo di calcolo:

- LISTAGG
- MEDIAN
- PERCENTILE_CONT
- PERCENTILE_DISC e APPROXIMATE PERCENTILE_DISC

Funzioni di aggregazione

Argomenti

- Funzione SUM
- Funzioni VAR_SAMP e VAR_POP

Le funzioni di aggregazione calcolano un singolo valore di risultato da un insieme di valori di input.

Le comunicazioni SELECT che utilizzano le funzioni di aggregazione possono comprendere due clausole facoltative: GROUP BY e HAVING. La sintassi per queste clausole è la seguente (usando la funzione COUNT come esempio):

```
SELECT count (*) expression FROM table_reference
WHERE condition [GROUP BY expression ] [ HAVING condition]
```

La clausola GROUP BY aggrega e raggruppa i risultati in base ai valori univoci in una colonna o colonne specificate. La clausola HAVING limita i risultati restituiti alle righe in cui una particolare condizione di aggregazione è vera, ad esempio il conteggio (*) > 1. La clausola HAVING viene utilizzata nello stesso modo di WHERE per limitare le righe in base al valore di una colonna. Per un esempio di queste clausole aggiuntive, consultare COUNT.

Le funzioni di aggregazione non accettano le funzioni di aggregazione nidificate o le funzioni finestra come argomenti.

Funzione ANY_VALUE

La funzione ANY_VALUE restituisce qualsiasi valore dai valori dell'espressione di input in modo non deterministico. Questa funzione restituisce NULL se l'espressione di input non determina la restituzione di righe. La funzione può restituire NULL anche se sono presenti valori NULL nell'espressione di input.

Sintassi

```
ANY_VALUE( [ DISTINCT | ALL ] expression )
```

Argomenti

DISTINCT | ALL

Specificare DISTINCT o ALL per restituire qualsiasi valore dai valori dell'espressione di input. L'argomento DISTINCT non ha alcun effetto e viene ignorato.

expression

L'espressione o la colonna di destinazione su cui viene eseguita la funzione. L'espressione è uno dei seguenti tipi di dati:

- SMALLINT
- INTEGER
- BIGINT
- DECIMAL
- REAL
- DOUBLE PRECISON
- BOOLEAN
- CHAR
- VARCHAR
- DATE
- TIMESTAMP
- TIMESTAMPTZ
- TIME
- TIMETZ
- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND
- VARBYTE
- SUPER
- HLLSKETCH
- GEOMETRY
- GEOGRAPHY

Valori restituiti

Restituisce lo stesso tipo di dati come espressione.

Note per l'utilizzo

Se un'istruzione che specifica la funzione ANY_VALUE per una colonna include anche un riferimento a una secondo colonna, la seconda colonna deve essere visualizzata in una clausola GROUP BY o inclusa in una funzione di aggregazione.

Esempi

Gli esempi utilizzano la tabella di eventi creata in Fase 4: Caricamento dei dati campione da Amazon S3 nella Guida alle operazioni di base di Amazon Redshift. Nell'esempio seguente viene restituita un'istanza di qualsiasi dateid in cui il nome evento è Eagles.

```
select any_value(dateid) as dateid, eventname from event where eventname ='Eagles'
 group by eventname;
```

Di seguito sono riportati i risultati.

```
dateid | eventname
-------+---------------
 1878  | Eagles
```

Nell'esempio seguente viene restituita un'istanza di qualsiasi dateid in cui il nome evento è Eagles o Cold War Kids.

```
select any_value(dateid) as dateid, eventname from event where eventname in('Eagles',
 'Cold War Kids') group by eventname;
```

Di seguito sono riportati i risultati.

```
dateid | eventname
-------+---------------
 1922  | Cold War Kids
 1878  | Eagles
```

Funzione APPROXIMATE PERCENTILE_DISC

APPROXIMATE PERCENTILE_DISC è una funzione di distribuzione inversa che presuppone un modello di distribuzione discreta. Prende un valore percentile e una specifica di ordinamento e

restituisce un elemento dall'insieme specificato. L'approssimazione consente alla funzione di eseguire molto più rapidamente, con un errore relativo basso di circa lo 0,5 percento.

Per un valore percentile dato, APPROXIMATE PERCENTILE_DISC utilizza un algoritmo di sintesi quantile per approssimare il percentile discreto dell'espressione nella clausola ORDER BY. APPROXIMATE PERCENTILE_DISC restituisce il valore con il valore di distribuzione cumulativa più piccolo (rispetto alla stessa specifica di ordinamento) maggiore o uguale al percentile.

APPROXIMATE PERCENTILE_DISC è una funzione solo del nodo di calcolo. La funzione restituisce un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift.

Sintassi

```
APPROXIMATE  PERCENTILE_DISC ( percentile )
WITHIN GROUP (ORDER BY expr)
```

Argomenti

percentile

Costante numerica compresa tra 0 e 1. I valori null vengono ignorati nel calcolo.

WITHIN GROUP (ORDER BY expr)

La clausola che specifica i valori numerici o di data/ora per ordinare e calcolare il percentile.

Valori restituiti

Lo stesso tipo di dati dell'espressione ORDER BY nella clausola WITHIN GROUP.

Note per l'utilizzo

Se l'affermazione APPROXIMATE PERCENTILE_DISC comprende una clausola GROUP BY, l'insieme di risultati è limitato. Il limite varia in base al tipo di nodo e al numero di nodi. Se il limite viene superato, la funzione ha esito negativo e restituisce il seguente errore.

```
GROUP BY limit for approximate percentile_disc exceeded.
```

Se è necessario valutare più gruppi di quelli consentiti dal limite, considerare l'utilizzo di Funzione PERCENTILE_CONT.

Esempi

L'esempio seguente restituisce il numero di vendite, le vendite totali e il valore del cinquantesimo percentile per le prime 10 date.

```
select top 10 date.caldate,
count(totalprice), sum(totalprice),
approximate percentile_disc(0.5)
within group (order by totalprice)
from listing
join date on listing.dateid = date.dateid
group by date.caldate
order by 3 desc;

caldate    | count | sum        | percentile_disc
-----------+-------+------------+----------------
2008-01-07 |   658 | 2081400.00 |         2020.00
2008-01-02 |   614 | 2064840.00 |         2178.00
2008-07-22 |   593 | 1994256.00 |         2214.00
2008-01-26 |   595 | 1993188.00 |         2272.00
2008-02-24 |   655 | 1975345.00 |         2070.00
2008-02-04 |   616 | 1972491.00 |         1995.00
2008-02-14 |   628 | 1971759.00 |         2184.00
2008-09-01 |   600 | 1944976.00 |         2100.00
2008-07-29 |   597 | 1944488.00 |         2106.00
2008-07-23 |   592 | 1943265.00 |         1974.00
```

Funzione AVG

La funzione AVG restituisce la media (media aritmetica) dei valori di espressione di input. La funzione AVG funziona con i valori numerici e ignora i valori NULL.

Sintassi

```
AVG ( [ DISTINCT | ALL ] expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. L'espressione è uno dei seguenti tipi di dati:

- SMALLINT
- INTEGER
- BIGINT
- NUMERIC
- DECIMAL
- REAL
- DOUBLE PRECISON
- SUPER

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati dall'espressione specificata prima di calcolare la media. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione per calcolare la media. ALL è il valore predefinito.

Tipi di dati

I tipi di argomenti supportati dalla funzione AVG sono SMALLINT, INTEGER, BIGINT, NUMERIC, DECIMAL, REAL, DOUBLE PRECISION e SUPER.

I tipi di restituzione supportati dalla funzione AVG sono:

- BIGINT per qualsiasi argomento di tipo integer
- DOUBLE PRECISION per un argomento del numero in virgola mobile
- Restituisce lo stesso tipo di dati come espressione per qualsiasi altro tipo di argomento.

La precisione di default per un risultato della funzione AVG con un argomento NUMERIC o DECIMAL è 38. Il ridimensionamento del risultato coincide con il ridimensionamento dell'argomento. Ad esempio, un AVG di una colonna DEC(5,2) restituisce un tipo di dati DEC(38,2).

Esempi

Trovare la quantità media venduta per transazione dalla tabella SALES:

```
select avg(qtysold)from sales;

avg
```

```
-----
2
(1 row)
```

Trovare il prezzo totale medio elencato per tutti gli elenchi:

```
select avg(numtickets*priceperticket) as avg_total_price from listing;

avg_total_price
-----------------
3034.41
(1 row)
```

Trovare il prezzo medio pagato, raggruppato per mese in ordine decrescente:

```
select avg(pricepaid) as avg_price, month
from sales, date
where sales.dateid = date.dateid
group by month
order by avg_price desc;

avg_price | month
-----------+-------
659.34 | MAR
655.06 | APR
645.82 | JAN
643.10 | MAY
642.72 | JUN
642.37 | SEP
640.72 | OCT
640.57 | DEC
635.34 | JUL
635.24 | FEB
634.24 | NOV
632.78 | AUG
(12 rows)
```

Funzione COUNT

La funzione COUNT conta le righe definite dall'espressione.

La funzione COUNT ha le seguenti variazioni.

- COUNT (*) conta tutte le righe nella tabella di destinazione indipendentemente dal fatto che includano valori null o meno.
- COUNT (espressione) calcola il numero di righe con valori non NULL in una colonna o espressione specifica.
- COUNT (DISTINCT espressione) calcola il numero di valori distinti non NULL in una colonna o espressione.
- APPROXIMATE COUNT DISTINCT esegue l'approssimazione del numero di valori distinti non NULL in una colonna o in un'espressione.

Sintassi

```
COUNT( * | expression )
```

```
COUNT ( [ DISTINCT | ALL ] expression )
```

```
APPROXIMATE COUNT ( DISTINCT expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. La funzione COUNT supporta tutti i tipi di dati degli argomenti.

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati dall'espressione specificata prima di eseguire il conteggio. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione per il conteggio. ALL è il valore predefinito.

APPROXIMATE

Se utilizzata con APPROSSIMATE, una funzione COUNT DISTINCT utilizza un HyperLogLog algoritmo per approssimare il numero di valori distinti non NULL in una colonna o in un'espressione. Le query che utilizzano la parola chiave APPROXIMATE eseguono molto più rapidamente, con un errore relativo basso di circa il 2%. L'approssimazione è garantita per le query che restituiscono un numero elevato di valori distinti, in milioni o più per query o per gruppo, se esiste una clausola group by (per gruppo). Per insiemi più piccoli di valori distinti, a migliaia,

l'approssimazione potrebbe essere più lenta di un conteggio preciso. APPROXIMATE può essere usata solo con COUNT DISTINCT.

Tipo restituito

La funzione COUNT restituisce BIGINT.

Esempi

Calcolare tutti gli utenti provenienti dallo stato della Florida:

```
select count(*) from users where state='FL';

count
-------
510
```

Calcolare tutti i nomi degli eventi dalla tabella EVENT:

```
select count(eventname) from event;

count
-------
8798
```

Calcolare tutti i nomi degli eventi dalla tabella EVENT:

```
select count(all eventname) from event;

count
-------
8798
```

Contare tutti gli ID di luogo univoci dalla tabella EVENT:

```
select count(distinct venueid) as venues from event;

venues
--------
204
```

Contare il numero di volte in cui ciascun venditore ha elencato lotti di oltre quattro biglietti in vendita. Raggruppare i risultati per ID venditore:

```
select count(*), sellerid from listing
where numtickets > 4
group by sellerid
order by 1 desc, 2;

count | sellerid
------+----------
12    |    6386
11    |    17304
11    |    20123
11    |    25428
...
```

I seguenti esempi mettono a confronto i valori di ritorno e i tempi di esecuzione per COUNT e APPROXIMATE COUNT.

```
select  count(distinct pricepaid) from sales;

count
-------
  4528


Time: 48.048 ms


select approximate count(distinct pricepaid) from sales;

count
-------
  4553


Time: 21.728 ms
```

Funzione LISTAGG

Per ogni gruppo in una query, la funzione di aggregazione LISTAGG ordina le righe di quel gruppo in base all'espressione ORDER BY, quindi concatena i valori in un'unica stringa.

LISTAGG è una funzione solo del nodo di calcolo. La funzione restituisce un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift. Per ulteriori informazioni, consulta Query sulle tabelle di catalogo.

Sintassi

```
LISTAGG( [DISTINCT] aggregate_expression [, 'delimiter' ] )
[ WITHIN GROUP (ORDER BY order_list) ]
```

Argomenti

DISTINCT

Una clausola che elimina i valori duplicati dall'espressione specificata prima della concatenazione. Gli spazi finali vengono ignorati. Ad esempio, le stringhe `'a'` e `'a '` vengono trattate come duplicati. LISTAGG usa il primo valore incontrato. Per ulteriori informazioni, consultare Significato degli spazi finali.

aggregate_expression

Qualsiasi espressione valida come il nome di una colonna che fornisce i valori da aggregare. I valori NULL e le stringhe vuote vengono ignorati.

delimiter

La costante di stringa per separare i valori concatenati. Il valore predefinito è NULL.

WITHIN GROUP (ORDER BY order_list)

Una clausola che specifica l'ordinamento dei valori aggregati.

Valori restituiti

VARCHAR(MAX). Se l'insieme dei risultati è maggiore della dimensione massima di VARCHAR, allora LISTAGG restituisce il seguente errore:

```
Invalid operation: Result size exceeds LISTAGG limit
```

Note per l'utilizzo

- Se un'affermazione comprende più funzioni LISTAGG che utilizzano le clausole WITHIN GROUP, ciascuna clausola WITHIN GROUP deve utilizzare gli stessi valori ORDER BY.

Ad esempio, la seguente istruzione restituisce un errore.

```
SELECT LISTAGG(sellerid)
WITHIN GROUP (ORDER BY dateid) AS sellers,
LISTAGG(dateid)
WITHIN GROUP (ORDER BY sellerid) AS dates
FROM sales;
```

La seguente istruzione viene eseguita correttamente.

```
SELECT LISTAGG(sellerid)
WITHIN GROUP (ORDER BY dateid) AS sellers,
LISTAGG(dateid)
WITHIN GROUP (ORDER BY dateid) AS dates
FROM sales;

SELECT LISTAGG(sellerid)
WITHIN GROUP (ORDER BY dateid) AS sellers,
LISTAGG(dateid) AS dates
FROM sales;
```

Esempi

L'esempio seguente aggrega gli ID venditore, ordinati per ID venditore.

```
SELECT LISTAGG(sellerid, ', ')
WITHIN GROUP (ORDER BY sellerid)
FROM sales
WHERE eventid = 4337;

listagg

-------------------------------------------------------------------------------------------------
380, 380, 1178, 1178, 1178, 2731, 8117, 12905, 32043, 32043, 32043, 32432, 32432,
 38669, 38750, 41498, 45676, 46324, 47188, 47188, 48294
```

Il seguente esempio utilizza DISTINCT per restituire un elenco di ID venditore univoci.

```
SELECT LISTAGG(DISTINCT sellerid, ', ')
WITHIN GROUP (ORDER BY sellerid)
```

```
FROM sales
WHERE eventid = 4337;

listagg

-----------------------------------------------------------------------------------------------
380, 1178, 2731, 8117, 12905, 32043, 32432, 38669, 38750, 41498, 45676, 46324, 47188,
 48294
```

L'esempio seguente aggrega gli ID venditore in ordine cronologico.

```
SELECT LISTAGG(sellerid, ', ')
WITHIN GROUP (ORDER BY dateid)
FROM sales
WHERE eventid = 4337;

   listagg
-----------------------------------------------------------------------------------------------
 41498, 47188, 47188, 1178, 1178, 1178, 380, 45676, 46324, 48294, 32043, 32043, 32432,
 12905, 8117, 38750, 2731, 32432, 32043, 380, 38669
```

Il seguente esempio restituisce un elenco separato dal carattere di barra verticale di date di vendita per l'acquirente con un ID di 660.

```
SELECT LISTAGG(
    (SELECT caldate FROM date WHERE date.dateid=sales.dateid), ' | '
)
WITHIN GROUP (ORDER BY sellerid DESC, salesid ASC)
FROM sales
WHERE buyerid = 660;

             listagg
-------------------------------------------------
2008-07-16 | 2008-07-09 | 2008-01-01 | 2008-10-26
```

Il seguente esempio restituisce un elenco separato da virgole di ID di vendita per gli ID acquirente 660, 661 e 662.

```
SELECT buyerid,
LISTAGG(salesid,', ')
WITHIN GROUP (ORDER BY salesid) AS sales_id
FROM sales
```

```
WHERE buyerid BETWEEN 660 AND 662
GROUP BY buyerid
ORDER BY buyerid;

buyerid |               sales_id
--------+-----------------------------------------------------
660     | 32872, 33095, 33514, 34548
661     | 19951, 20517, 21695, 21931
662     | 3318, 3823, 4215, 51980, 53202, 55908, 57832, 171603
```

Funzione MAX

La funzione MAX restituisce il valore massimo in un insieme di righe. È possibile utilizzare DISTINCT oppure ALL ma non influenzano il risultato.

Sintassi

```
MAX ( [ DISTINCT | ALL ] expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. L'espressione è uno dei seguenti tipi di dati:

- SMALLINT
- INTEGER
- BIGINT
- DECIMAL
- REAL
- DOUBLE PRECISON
- CHAR
- VARCHAR
- DATE
- TIMESTAMP
- TIMESTAMPTZ

- TIME
- TIMETZ
- VARBYTE
- SUPER

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati dall'espressione specificata prima di calcolare il massimo. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione per calcolare il massimo. ALL è il valore predefinito.

Tipi di dati

Restituisce lo stesso tipo di dati come espressione. L'equivalente booleano della funzione MIN è il Funzione BOOL_AND, e l'equivalente booleano di MAX è il Funzione BOOL_OR.

Esempi

Trovare il prezzo più alto pagato da tutte le vendite:

```
select max(pricepaid) from sales;

max
----------
12624.00
(1 row)
```

Trovare il prezzo più alto pagato per biglietto da tutte le vendite:

```
select max(pricepaid/qtysold) as max_ticket_price
from sales;

max_ticket_price
-----------------
2500.00000000
(1 row)
```

Funzione MEDIAN

Calcola il valore mediano per l'intervallo di valori. I valori NULL nell'intervallo vengono ignorati.

MEDIAN è una funzione di distribuzione inversa che presuppone un modello di distribuzione continua.

MEDIAN è un caso speciale di PERCENTILE_CONT.

MEDIAN è una funzione solo del nodo di calcolo. La funzione restituisce un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift.

Sintassi

```
MEDIAN(median_expression)
```

Argomenti

median_expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

Tipi di dati

Il tipo di ritorno è determinato dal tipo di dati di median_expression. La seguente tabella mostra il tipo di restituzione per ciascun median_expression tipo di dati.

Input type (Tipo input)	Tipo restituito
INT2, INT4, INT8, NUMERIC, DECIMAL	DECIMAL
FLOAT, DOUBLE	DOUBLE
DATE	DATE
TIMESTAMP	TIMESTAMP
TIMESTAMPTZ	TIMESTAMPTZ

Note per l'utilizzo

Se l'argomento median_expression è un tipo di dati DECIMAL definito con la precisione massima di 38 cifre, è possibile che MEDIAN restituisca un risultato inaccurato o un errore. Se il valore di

ritorno della funzione MEDIAN supera le 38 cifre, il risultato viene troncato per adattarsi, il che causa una perdita della precisione. Se, durante l'interpolazione, un risultato intermedio supera la precisione massima, si verifica un'eccedenza numerica e la funzione restituisce un errore. Per evitare queste condizioni, consigliamo di utilizzare un tipo di dati con una precisione inferiore o di assegnare l'argomento median_expression a una precisione inferiore.

Se un'istruzione comprende più chiamate a funzioni di aggregazione basate su ordinamento (LISTAGG, PERCENTILE_CONT o MEDIAN), devono utilizzare tutti gli stessi valori ORDER BY. Notare che MEDIAN applica un ordine implicito sul valore dell'espressione.

Ad esempio, la seguente istruzione restituisce un errore.

```
SELECT TOP 10 salesid, SUM(pricepaid),
PERCENTILE_CONT(0.6) WITHIN GROUP(ORDER BY salesid),
MEDIAN(pricepaid)
FROM sales
GROUP BY salesid, pricepaid;

An error occurred when executing the SQL command:
SELECT TOP 10 salesid, SUM(pricepaid),
PERCENTILE_CONT(0.6) WITHIN GROUP(ORDER BY salesid),
MEDIAN(pricepaid)
FROM sales
GROUP BY salesid, pricepaid;

ERROR: within group ORDER BY clauses for aggregate functions must be the same
```

La seguente istruzione viene eseguita normalmente.

```
SELECT TOP 10 salesid, SUM(pricepaid),
PERCENTILE_CONT(0.6) WITHIN GROUP(ORDER BY salesid),
MEDIAN(salesid)
FROM sales
GROUP BY salesid, pricepaid;
```

Esempi

Gli esempi seguenti utilizzano il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

L'esempio seguente mostra che MEDIAN produce gli stessi risultati di PERCENTILE_CONT(0.5).

```
SELECT TOP 10 DISTINCT sellerid, qtysold,
PERCENTILE_CONT(0.5) WITHIN GROUP(ORDER BY qtysold),
MEDIAN(qtysold)
FROM sales
GROUP BY sellerid, qtysold;

+----------+---------+-----------------+--------+
| sellerid | qtysold | percentile_cont | median |
+----------+---------+-----------------+--------+
|        2 |       2 |               2 |      2 |
|       26 |       1 |               1 |      1 |
|       33 |       1 |               1 |      1 |
|       38 |       1 |               1 |      1 |
|       43 |       1 |               1 |      1 |
|       48 |       2 |               2 |      2 |
|       48 |       3 |               3 |      3 |
|       77 |       4 |               4 |      4 |
|       85 |       4 |               4 |      4 |
|       95 |       2 |               2 |      2 |
+----------+---------+-----------------+--------+
```

L'esempio seguente trova la quantità media venduta per ogni sellerid.

```
SELECT sellerid,
MEDIAN(qtysold)
FROM sales
GROUP BY sellerid
ORDER BY sellerid
LIMIT 10;

+----------+--------+
| sellerid | median |
+----------+--------+
|        1 |    1.5 |
|        2 |      2 |
|        3 |      2 |
|        4 |      2 |
|        5 |      1 |
|        6 |      1 |
|        7 |    1.5 |
|        8 |      1 |
|        9 |      4 |
|       12 |      2 |
```

```
+----------+---------+
```

Per verificare i risultati della query precedente per il primo sellerid, utilizza l'esempio seguente.

```
SELECT qtysold
FROM sales
WHERE sellerid=1;

+---------+
| qtysold |
+---------+
|       2 |
|       1 |
+---------+
```

Funzione MIN

La funzione MIN restituisce il valore minimo in un insieme di righe. È possibile utilizzare DISTINCT oppure ALL ma non influenzano il risultato.

Sintassi

```
MIN ( [ DISTINCT | ALL ] expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. L'espressione è uno dei seguenti tipi di dati:

- SMALLINT
- INTEGER
- BIGINT
- DECIMAL
- REAL
- DOUBLE PRECISON
- CHAR

- VARCHAR
- DATE
- TIMESTAMP
- TIMESTAMPTZ
- TIME
- TIMETZ
- VARBYTE
- SUPER

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati dall'espressione specificata prima di calcolare il minimo. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione per calcolare il minimo. ALL è il valore predefinito.

Tipi di dati

Restituisce lo stesso tipo di dati come espressione. L'equivalente booleano della funzione MIN è Funzione BOOL_AND, e l'equivalente booleano di MAX è Funzione BOOL_OR.

Esempi

Trovare il prezzo più basso pagato da tutte le vendite:

```
select min(pricepaid) from sales;

min
-------
20.00
(1 row)
```

Trovare il prezzo più basso pagato per biglietto da tutte le vendite:

```
select min(pricepaid/qtysold)as min_ticket_price
from sales;

min_ticket_price
------------------
20.00000000
```

```
(1 row)
```

Funzione PERCENTILE_CONT

PERCENTILE_CONT è una funzione di distribuzione inversa che presuppone un modello di distribuzione continua. Prende un valore percentile e una specifica di ordinamento e restituisce un valore interpolato che rientrerebbe nel valore percentile dato rispetto alla specifica di ordinamento.

PERCENTILE_CONT calcola un'interpolazione lineare tra i valori dopo averli ordinati. Usando il valore percentile `(P)` e il numero di righe non null `(N)` nel gruppo di aggregazione, la funzione calcola il numero di righe dopo aver ordinato le righe secondo la specifica di ordinamento. Questo numero di riga `(RN)` è calcolato secondo la formula `RN = (1+ (P*(N-1))`. Il risultato finale della funzione di aggregazione è calcolato mediante interpolazione lineare tra i valori delle righe ai numeri di riga `CRN = CEILING(RN)` e `FRN = FLOOR(RN)`.

Il risultato finale sarà il seguente.

Se `(CRN = FRN = RN)` allora il risultato è `(value of expression from row at RN)`

Altrimenti il risultato è il seguente:

`(CRN - RN) * (value of expression for row at FRN) + (RN - FRN) * (value of expression for row at CRN)`.

PERCENTILE_CONT è una funzione solo del nodo di calcolo. La funzione restituisce un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift.

Sintassi

```
PERCENTILE_CONT(percentile)
WITHIN GROUP(ORDER BY expr)
```

Argomenti

percentile

Costante numerica tra 0 e 1. I valori `NULL` vengono ignorati nel calcolo.

expr

Specifica i valori numerici o di data/ora per ordinare e calcolare il percentile.

Valori restituiti

Il tipo di restituzione è determinato dal tipo di dati dell'espressione ORDER BY nella clausola WITHIN GROUP. La seguente tabella mostra il tipo di restituzione per ciascun tipo di dati dell'espressione ORDER BY.

Input type (Tipo input)	Tipo restituito
INT2, INT4, INT8, NUMERIC, DECIMAL	DECIMAL
FLOAT, DOUBLE	DOUBLE
DATE	DATE
TIMESTAMP	TIMESTAMP
TIMESTAMPTZ	TIMESTAMPTZ

Note per l'utilizzo

Se l'espressione ORDER BY è un tipo di dati DECIMAL definito con la precisione massima di 38 cifre, è possibile che PERCENTILE_CONT restituirà un risultato inaccurato o un errore. Se il valore di ritorno della funzione PERCENTILE_CONT supera le 38 cifre, il risultato viene troncato per adattarsi, il che causa una perdita della precisione. Se, durante l'interpolazione, un risultato intermedio supera la precisione massima, si verifica un'eccedenza numerica e la funzione restituisce un errore. Per evitare queste condizioni, consigliamo di utilizzare un tipo di dati con una precisione inferiore o di assegnare l'espressione ORDER BY a una precisione inferiore.

Se un'istruzione comprende più chiamate a funzioni di aggregazione basate su ordinamento (LISTAGG, PERCENTILE_CONT o MEDIAN), devono utilizzare tutti gli stessi valori ORDER BY. Notare che MEDIAN applica un ordine implicito sul valore dell'espressione.

Ad esempio, la seguente istruzione restituisce un errore.

```
SELECT TOP 10 salesid, SUM(pricepaid),
PERCENTILE_CONT(0.6) WITHIN GROUP(ORDER BY salesid),
MEDIAN(pricepaid)
FROM sales
GROUP BY salesid, pricepaid;
```

```
An error occurred when executing the SQL command:
SELECT TOP 10 salesid, SUM(pricepaid),
PERCENTILE_CONT(0.6) WITHIN GROUP(ORDER BY salesid),
MEDIAN(pricepaid)
FROM sales
GROUP BY salesid, pricepaid;

ERROR: within group ORDER BY clauses for aggregate functions must be the same
```

La seguente istruzione viene eseguita normalmente.

```
SELECT TOP 10 salesid, SUM(pricepaid),
PERCENTILE_CONT(0.6) WITHIN GROUP(ORDER BY salesid),
MEDIAN(salesid)
FROM sales
GROUP BY salesid, pricepaid;
```

Esempi

Gli esempi seguenti utilizzano il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

L'esempio seguente mostra che PERCENTILE_CONT(0.5) produce gli stessi risultati di MEDIAN.

```
SELECT TOP 10 DISTINCT sellerid, qtysold,
PERCENTILE_CONT(0.5) WITHIN GROUP(ORDER BY qtysold),
MEDIAN(qtysold)
FROM sales
GROUP BY sellerid, qtysold;

+----------+---------+-----------------+--------+
| sellerid | qtysold | percentile_cont | median |
+----------+---------+-----------------+--------+
|        2 |       2 |               2 |      2 |
|       26 |       1 |               1 |      1 |
|       33 |       1 |               1 |      1 |
|       38 |       1 |               1 |      1 |
|       43 |       1 |               1 |      1 |
|       48 |       2 |               2 |      2 |
|       48 |       3 |               3 |      3 |
|       77 |       4 |               4 |      4 |
```

```
|       85 |       4 |                4 |      4 |
|       95 |       2 |                2 |      2 |
+----------+---------+------------------+--------+
```

L'esempio seguente trova PERCENTILE_CONT(0.5) e PERCENTILE_CONT(0.75) per la quantità venduta per ogni sellerid nella tabella SALES.

```
SELECT sellerid,
PERCENTILE_CONT(0.5) WITHIN GROUP(ORDER BY qtysold) as pct_05,
PERCENTILE_CONT(0.75) WITHIN GROUP(ORDER BY qtysold) as pct_075
FROM sales
GROUP BY sellerid
ORDER BY sellerid
LIMIT 10;

+----------+--------+---------+
| sellerid | pct_05 | pct_075 |
+----------+--------+---------+
|        1 |    1.5 |    1.75 |
|        2 |      2 |    2.25 |
|        3 |      2 |       3 |
|        4 |      2 |       2 |
|        5 |      1 |     1.5 |
|        6 |      1 |       1 |
|        7 |    1.5 |    1.75 |
|        8 |      1 |       1 |
|        9 |      4 |       4 |
|       12 |      2 |    3.25 |
+----------+--------+---------+
```

Per verificare i risultati della query precedente per il primo sellerid, utilizza l'esempio seguente.

```
SELECT qtysold
FROM sales
WHERE sellerid=1;

+---------+
| qtysold |
+---------+
|       2 |
|       1 |
+---------+
```

Funzioni STDDEV_SAMP e STDDEV_POP

Le funzioni STDDEV_SAMP e STDDEV_POP restituiscono la deviazione standard del campione e della popolazione di un insieme di valori numerici (integer, numero decimale, numero in virgola mobile). Il risultato della funzione STDDEV_SAMP è equivalente alla radice quadrata della varianza campione dello stesso insieme di valori.

STDDEV_SAMP e STDDEV sono sinonimi della stessa funzione.

Sintassi

```
STDDEV_SAMP | STDDEV ( [ DISTINCT | ALL ] expression)
STDDEV_POP ( [ DISTINCT | ALL ] expression)
```

L'espressione deve avere un tipo di dati integer, numero decimale o numero in virgola mobile. Indipendentemente dal tipo di dati dell'espressione, il tipo di restituzione di questa funzione è un numero a precisione doppia.

> **Note**
>
> La deviazione standard viene calcolata utilizzando l'aritmetica del numero in virgola mobile, che potrebbe causare una leggera imprecisione.

Note per l'utilizzo

Quando la deviazione standard del campione (STDDEV o STDDEV_SAMP) viene calcolata per un'espressione che consiste in un singolo valore, il risultato della funzione è NULL non 0.

Esempi

La seguente query restituisce la media dei valori nella colonna VENUESEATS della tabella VENUE, seguita dalla deviazione standard del campione e dalla deviazione standard della popolazione dello stesso insieme di valori. VENUESEATS è una colonna INTEGER. Il ridimensionamento del risultato è ridotto a 2 cifre.

```
select avg(venueseats),
cast(stddev_samp(venueseats) as dec(14,2)) stddevsamp,
cast(stddev_pop(venueseats) as dec(14,2)) stddevpop
```

```
from venue;

avg  | stddevsamp | stddevpop
-------+------------+-----------
17503 |   27847.76 |  27773.20
(1 row)
```

La seguente query restituisce la deviazione standard del campione per la colonna COMMISSIONE nella tabella SALES. COMMISSION è una colonna DECIMAL. Il ridimensionamento del risultato è ridotto a 10 cifre.

```
select cast(stddev(commission) as dec(18,10))
from sales;

stddev
----------------
130.3912659086
(1 row)
```

La seguente query assegna la deviazione standard del campione per la colonna COMMISSIONE come un integer.

```
select cast(stddev(commission) as integer)
from sales;

stddev
--------
130
(1 row)
```

La seguente query restituisce sia la deviazione standard del campione sia la radice quadrata della varianza campionaria per la colonna COMMISSIONE. I risultati di questi calcoli sono gli stessi.

```
select
cast(stddev_samp(commission) as dec(18,10)) stddevsamp,
cast(sqrt(var_samp(commission)) as dec(18,10)) sqrtvarsamp
from sales;

stddevsamp   |  sqrtvarsamp
----------------+----------------
130.3912659086 | 130.3912659086
```

```
(1 row)
```

Funzione SUM

La funzione SUM restituisce la somma dei valori dell'espressione o della colonna di input. La funzione SUM funziona con i valori numerici e ignora i valori NULL.

Sintassi

```
SUM ( [ DISTINCT | ALL ] expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. L'espressione è uno dei seguenti tipi di dati:

- SMALLINT
- INTEGER
- BIGINT
- NUMERIC
- DECIMAL
- REAL
- DOUBLE PRECISON
- SUPER

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati dall'espressione specificata prima di calcolare la somma. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione per calcolare la somma. ALL è il valore predefinito.

Tipi di dati

I tipi di argomenti supportati dalla funzione SUM sono SMALLINT, INTEGER, BIGINT, NUMERIC, DECIMAL, REAL, DOUBLE PRECISION e SUPER.

I tipi di restituzione supportati dalla funzione SUM sono

- BIGINT per gli argomenti BIGINT, SMALLINT, e INTEGER
- NUMERIC per gli argomenti NUMERIC
- DOUBLE PRECISION per argomenti del numero in virgola mobile
- Restituisce lo stesso tipo di dati come espressione per qualsiasi altro tipo di argomento.

La precisione di default per un risultato della funzione SUM con un argomento NUMERIC o DECIMAL è 38. Il ridimensionamento del risultato coincide con il ridimensionamento dell'argomento. Ad esempio, un SUM di una colonna DEC(5,2) restituisce un tipo di dati DEC(38,2).

Esempi

Trovare la somma di tutte le commissioni pagate dalla tabella SALES:

```
select sum(commission) from sales;

sum
-------------
16614814.65
(1 row)
```

Trovare il numero di posti in tutti i luoghi dello stato della Florida:

```
select sum(venueseats) from venue
where venuestate = 'FL';

sum
--------
250411
(1 row)
```

Trovare il numero di posti venduti a maggio:

```
select sum(qtysold) from sales, date
where sales.dateid = date.dateid and date.month = 'MAY';

sum
-------
32291
```

```
(1 row)
```

Funzioni VAR_SAMP e VAR_POP

Le funzioni VAR_SAMP e VAR_POP restituiscono la varianza del campione e della popolazione di un insieme di valori numerici (integer, numero decimale, numero in virgola mobile). Il risultato della funzione VAR_SAMP è equivalente alla deviazione standard del campione quadrato dello stesso insieme di valori.

VAR_SAMP e VARIANCE sono sinonimi della stessa funzione.

Sintassi

```
VAR_SAMP | VARIANCE ( [ DISTINCT | ALL ] expression)
VAR_POP ( [ DISTINCT | ALL ] expression)
```

L'espressione deve avere un tipo di dati integer, numero decimale o numero in virgola mobile. Indipendentemente dal tipo di dati dell'espressione, il tipo di restituzione di questa funzione è un numero a precisione doppia.

> **Note**
>
> I risultati di queste funzioni potrebbero variare tra i cluster di data warehouse, a seconda della configurazione del cluster in ciascun caso.

Note per l'utilizzo

Quando la varianza del campione (VARIANCE o VAR_SAMP) viene calcolata per un'espressione che consiste in un singolo valore, il risultato della funzione è NULL non 0.

Esempi

La seguente query restituisce l'esempio arrotondato e la varianza di popolazione della colonna NUMTICKETS nella tabella LISTING.

```
select avg(numtickets),
round(var_samp(numtickets)) varsamp,
round(var_pop(numtickets)) varpop
from listing;
```

```
avg | varsamp | varpop
-----+---------+--------
10 |      54 |     54
(1 row)
```

La seguente query esegue gli stessi calcoli ma assegna i risultati ai valori decimali.

```
select avg(numtickets),
cast(var_samp(numtickets) as dec(10,4)) varsamp,
cast(var_pop(numtickets) as dec(10,4)) varpop
from listing;

avg | varsamp | varpop
-----+---------+---------
10 | 53.6291 | 53.6288
(1 row)
```

Funzioni dell'array

Di seguito, è riportata una descrizione delle funzioni di array per SQL che Amazon Redshift supporta per accedere e manipolare gli array.

Argomenti

- Funzione array
- funzione array_concat
- Funzione array_flatten
- Funzione get_array_length
- Funzione split_to_array
- Funzione subarray

Funzione array

Crea un array del tipo di dati SUPER.

Sintassi

```
ARRAY( [ expr1 ] [ , expr2 [ , ... ] ] )
```

Argomento

expr1, expr2

Espressioni di qualsiasi tipo di dati Amazon Redshift eccetto i tipi di data e ora, poiché Amazon Redshift non esegue il cast dei tipi di data e ora nel tipo di dati SUPER. Non è necessario che gli argomenti siano dello stesso tipo di dati.

Tipo restituito

La funzione array restituisce il tipo di dati SUPER.

Esempio

Gli esempi seguenti mostrano un array di valori numerici e un array di tipi di dati diversi.

```
--an array of numeric values
select array(1,50,null,100);
      array
------------------
 [1,50,null,100]
(1 row)

--an array of different data types
select array(1,'abc',true,3.14);
        array
----------------------
 [1,"abc",true,3.14]
(1 row)
```

funzione array_concat

La funzione array_concat concatena due array per creare un array contenente tutti gli elementi nel primo array seguiti da tutti gli elementi nel secondo array. I due argomenti devono essere array validi.

Sintassi

```
array_concat( super_expr1,  super_expr2 )
```

Argomenti

super_expr1

Il valore che specifica il primo dei due array da concatenare.

super_expr2

Il valore che specifica il secondo dei due array da concatenare.

Tipo restituito

La funzione array_concat restituisce un valore di dati SUPER.

Esempio

Gli esempi seguenti mostrano la concatenazione di due array dello stesso tipo e la concatenazione di due array di tipi diversi.

```
-- concatenating two arrays
SELECT ARRAY_CONCAT(ARRAY(10001,10002),ARRAY(10003,10004));
              array_concat
-----------------------------------
 [10001,10002,10003,10004]
(1 row)

-- concatenating two arrays of different types
SELECT ARRAY_CONCAT(ARRAY(10001,10002),ARRAY('ab','cd'));
          array_concat
------------------------------
 [10001,10002,"ab","cd"]
(1 row)
```

Funzione array_flatten

Unisce più array in un solo array di tipo SUPER.

Sintassi

```
array_flatten( super_expr1,super_expr2,.. )
```

Argomenti

super_expr1,super_expr2

Una espressione SUPER valida della forma di array.

Tipo restituito

La funzione array_flatten restituisce un valore di dati SUPER.

Esempio

Nel seguente esempio viene mostrata una funzione array_flatten.

```
SELECT ARRAY_FLATTEN(ARRAY(ARRAY(1,2,3,4),ARRAY(5,6,7,8),ARRAY(9,10)));
     array_flatten
-----------------------
 [1,2,3,4,5,6,7,8,9,10]
(1 row)
```

Funzione get_array_length

Restituisce la lunghezza dell'array specificato. La funzione GET_ARRAY_LENGTH restituisce la lunghezza di un array SUPER dato un percorso oggetto o array.

Sintassi

```
get_array_length( super_expr )
```

Argomenti

super_expr

Una espressione SUPER valida della forma di array.

Tipo restituito

La funzione get_array_length restituisce un BIGINT.

Esempio

L'esempio seguente mostra una funzione get_array_length.

```
SELECT GET_ARRAY_LENGTH(ARRAY(1,2,3,4,5,6,7,8,9,10));
 get_array_length
----------------------
            10
(1 row)
```

Funzione split_to_array

Utilizza un delimitatore come parametro facoltativo. Se non è presente alcun delimitatore, il valore di default è una virgola.

Sintassi

```
split_to_array( string,delimiter )
```

Argomenti

stringa

La stringa di input da dividere.

delimiter

Un valore facoltativo su cui verrà divisa la stringa di input. L'impostazione predefinita è una virgola.

Tipo restituito

La funzione split_to_array restituisce un valore di dati SUPER.

Esempio

L'esempio seguente mostra una funzione split_to_array.

```
SELECT SPLIT_TO_ARRAY('12|345|6789', '|');
     split_to_array
------------------------
 ["12","345","6789"]
(1 row)
```

Funzione subarray

Manipola gli array per restituire un sottoinsieme degli array di input.

Sintassi

```
SUBARRAY( super_expr, start_position, length )
```

Argomenti

super_expr

Una espressione SUPER valida in forma di array.

start_position

La posizione all'interno dell'array per iniziare l'estrazione, a partire dalla posizione indice 0. Una posizione negativa viene conteggiata all'indietro dalla fine della matrice.

length

Il numero di elementi da estrarre (la lunghezza della sottostringa).

Tipo restituito

La funzione subarray restituisce un valore di dati SUPER.

Esempi

Di seguito è riportato un esempio di una funzione subarray:

```
 SELECT SUBARRAY(ARRAY('a', 'b', 'c', 'd', 'e', 'f'), 2, 3);
   subarray
---------------
 ["c","d","e"]
(1 row)
```

Funzioni di aggregazione bit per bit

Le funzioni di aggregazione bit per bit calcolano le operazioni di bit per eseguire l'aggregazione di colonne intere e colonne che possono essere convertite o arrotondate a valori interi.

Argomenti

Utilizzo di NULL nelle aggregazioni bit-wise

Quando viene applicata una funzione bit-wise a una colonna che è nullable, qualsiasi valore NULL viene eliminato prima che venga calcolato il risultato della funzione. Se nessuna riga ha i requisiti per l'aggregazione, la funzione bit-wise restituisce NULL. Lo stesso comportamento si applica alle normali funzioni di aggregazione. Di seguito è riportato un esempio.

```
select sum(venueseats), bit_and(venueseats) from venue
where venueseats is null;

sum  | bit_and
------+---------
null |    null
(1 row)
```

Supporto DISTINCT per le aggregazioni bit per bit

Come altre funzioni di aggregazione, le funzioni bit-wise supportano la parola chiave DISTINCT.

Tuttavia, l'utilizzo di DISTINCT con queste funzioni non ha alcun impatto sui risultati. La prima istanza di un valore è sufficiente per soddisfare le operazioni AND o OR bit-wise. Non fa alcuna differenza se sono presenti valori duplicati nell'espressione valutata.

Poiché è probabile che l'elaborazione DISTINCT generi un sovraccarico nell'esecuzione di alcune query, non utilizzare DISTINCT con le funzioni bit-wise.

Esempi di panoramica per le funzioni bit-wise

Di seguito, è possibile trovare alcuni esempi di panoramica che dimostrano come utilizzare le funzioni bit-wise. Sono presenti anche esempi di codice specifici con la descrizione di ogni funzione.

Gli esempi delle funzioni bit-wise si basano sul database di esempio TICKIT. La tabella USERS nel database di esempio TICKIT contiene diverse colonne booleane che indicano se a ciascun utente sono noti tipi diversi di eventi, come ad esempio sport, teatro, opera e così via. Di seguito è riportato un esempio.

```
select userid, username, lastname, city, state,
likesports, liketheatre
from users limit 10;

userid | username | lastname  |     city     | state | likesports | liketheatre
-------+----------+-----------+--------------+-------+------------+-------------
1 | JSG99FHE | Taylor    | Kent         | WA    | t          | t
9 | MSD36KVR | Watkins   | Port Orford  | MD    | t          | f
```

Si supponga che una nuova versione della tabella USERS sia costruita in un modo diverso. In questa nuova versione, una singola colonna di numeri interi che definisce (in formato binario) otto tipi di eventi che piace o non piace a ciascun utente. In questo progetto, ogni posizione di bit rappresenta un tipo di evento. Un utente a cui piacciono tutti gli otto tipi ha tutti gli otto bit impostati su 1 (come nella prima riga della seguente tabella). Un utente a cui non piace uno di questi eventi ha tutti gli otto bit impostati su 0 (vedere la seconda riga). Un utente che ama solo sport e jazz è rappresentato nella terza riga:

	SPORTS	THEATRE	JAZZ	OPERA	ROCK	VEGAS	BROADW	CLASSICAL
Utente 1	1	1	1	1	1	1	1	1
Utente 2	0	0	0	0	0	0	0	0
Utente 3	1	0	1	0	0	0	0	0

Nella tabella del database, questi valori binari potrebbero essere archiviati in una singola colonna LIKES come numeri interi:

Utente	Valore binario	Valore archiviato (intero)
Utente 1	11111111	255
Utente 2	00000000	0
Utente 3	10100000	160

Funzione BIT_AND

La funzione BIT_AND esegue operazioni AND bit per bit su tutti i valori di una singola colonna o espressione intera. Questa funzione aggrega ogni bit di ciascun valore binario che corrisponde a ciascun valore intero nell'espressione.

La funzione BIT_AND restituisce un risultato di `0` se nessuno dei bit è impostato su 1 in tutti i valori. Se uno o più bit è impostato su 1 in tutti i valori, la funzione restituisce un valore intero. Questo intero è il numero che corrisponde al valore binario per quei bit.

Ad esempio, una tabella contiene quattro valori interni in una colonna: 3, 7, 10, e 22. Questi numeri interi sono rappresentati in forma binaria nel seguente modo:

Numero intero	Valore binario
3	11
7	111
10	1010
22	10110

Un'operazione BIT_AND su questo set di dati rileva che tutti i bit sono impostati solo nella posizione. 1 second-to-last Il risultato è un valore binario di `00000010`, che rappresenta il valore intero 2. Di conseguenza, la funzione BIT_AND restituisce 2.

Sintassi

```
BIT_AND ( [DISTINCT | ALL] expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. Questa espressione deve avere un tipo di dati INT, INT2 o INT8. La funzione restituisce un tipo di dati INT, INT2 o INT8 equivalente.

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati per l'espressione specificata prima di calcolare il risultato. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati. ALL è il valore predefinito. Per ulteriori informazioni, consultare Supporto DISTINCT per le aggregazioni bit per bit.

Esempi

Dato che le informazioni aziendali significative sono memorizzate in colonne di numeri interi, è possibile utilizzare le funzioni bit-wise per estrarre e aggregare tali informazioni. La seguente query applica la funzione BIT_AND alla colonna LIKES in una tabella denominata USERLIKES e raggruppa i risultati in base alla colonna CITY.

```
select city, bit_and(likes) from userlikes group by city
order by city;
city          | bit_and
--------------+---------
Los Angeles   |       0
Sacramento    |       0
San Francisco |       0
San Jose      |      64
Santa Barbara |     192
(5 rows)
```

Questi risultati possono essere interpretati nel modo seguente:

- Il valore intero 192 per Santa Barbara si traduce nel valore binario `11000000`. In altre parole, a tutti gli utenti di questa città piacciono lo sport e il teatro, ma non a tutti gli utenti piacciono altri tipi di evento.
- Il numero intero 64 si traduce in `01000000`. Pertanto, per gli utenti a San Jose, l'unico tipo di evento che piace a tutti è il teatro.

- I valori di `0` per le altre tre città indica che nessun "like" di gradimento è condiviso da tutti gli utenti in quelle città.

Funzione BIT_OR

La funzione BIT_OR esegue operazioni OR bit per bit su tutti i valori di una singola colonna o espressione intera. Questa funzione aggrega ogni bit di ciascun valore binario che corrisponde a ciascun valore intero nell'espressione.

Ad esempio, si supponga che la propria tabella contenga quattro valori interi in una colonna: 3, 7, 10, e 22. Questi numeri interi sono rappresentati in forma binaria nel seguente modo:

Numero intero	Valore binario
3	11
7	111
10	1010
22	10110

Se si applica la funzione BIT_OR all'insieme di valori interi, l'operazione cerca qualsiasi valore in cui 1 è presente in ciascuna posizione. In questo caso, un 1 esiste nelle ultime cinque posizioni per almeno uno dei valori, ottenendo un risultato binario di `00011111`; pertanto, la funzione restituisce `31` (oppure `16 + 8 + 4 + 2 + 1`).

Sintassi

```
BIT_OR ( [DISTINCT | ALL] expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. Questa espressione deve avere un tipo di dati INT, INT2 o INT8. La funzione restituisce un tipo di dati INT, INT2 o INT8 equivalente.

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati per l'espressione specificata prima di calcolare il risultato. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati. ALL è il valore predefinito. Per ulteriori informazioni, consultare Supporto DISTINCT per le aggregazioni bit per bit.

Esempio

La seguente query applica la funzione BIT_OR alla colonna LIKES in una tabella denominata USERLIKES e raggruppa i risultati in base alla colonna CITY.

```
select city, bit_or(likes) from userlikes group by city
order by city;
city          | bit_or
--------------+--------
Los Angeles   |    127
Sacramento    |    255
San Francisco |    255
San Jose      |    255
Santa Barbara |    255
(5 rows)
```

Per quattro delle città elencate, tutti i tipi di evento piacciono ad almeno un utente (`255=11111111`). Per Los Angeles, tutti i tipi di evento fatta eccezione per gli sport piacciono ad almeno un utente (`127=01111111`).

Funzione BOOL_AND

La funzione BOOL_AND opera su una singola colonna o espressione booleana o intera. Questa funzione applica una logica simile alle funzioni BIT_AND e BIT_OR. Per questa funzione, il tipo restituito è un valore booleano (`true` o `false`).

Se tutti i valori in un insieme sono veri, viene restituita la funzione BOOL_AND `true` (`t`). Se un valore è falso, la funzione restituisce `false` (`f`).

Sintassi

```
BOOL_AND ( [DISTINCT | ALL] expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. Questa espressione deve avere un tipo di dati intero o BOOLEAN. Il tipo restituito della funzione è BOOLEAN.

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati per l'espressione specificata prima di calcolare il risultato. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati. ALL è il valore predefinito. Per ulteriori informazioni, consultare Supporto DISTINCT per le aggregazioni bit per bit.

Esempi

È possibile utilizzare le funzioni booleane rispetto alle espressioni booleane o alle espressioni intere. Ad esempio, la seguente query restituisce i risultati dalla tabella USERS standard nel database TICKIT, che ha diverse colonne booleane.

La funzione BOOL_AND restituisce `false` per tutte e cinque le righe. Non a tutti gli utenti in ciascuno di questi stati piace lo sport.

```
select state, bool_and(likesports) from users
group by state order by state limit 5;

state | bool_and
------+---------
AB    | f
AK    | f
AL    | f
AZ    | f
BC    | f
(5 rows)
```

Funzione BOOL_OR

La funzione BOOL_OR opera su una singola colonna o espressione booleana o intera. Questa funzione applica una logica simile alle funzioni BIT_AND e BIT_OR. Per questa funzione, il tipo restituito è un valore booleano (`true`, `false` o NULL).

Se uno o più valori in un set lo sonotrue, la funzione BOOL_OR restituisce (). `true` `t` Se tutti i valori di un set sonofalse, la funzione restituisce `false` (). `f` NULL può essere restituito se il valore è sconosciuto.

Sintassi

```
BOOL_OR ( [DISTINCT | ALL] expression )
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione. Questa espressione deve avere un tipo di dati intero o BOOLEAN. Il tipo restituito della funzione è BOOLEAN.

DISTINCT | ALL

Con l'argomento DISTINCT, la funzione elimina tutti i valori duplicati per l'espressione specificata prima di calcolare il risultato. Con l'argomento ALL, la funzione mantiene tutti i valori duplicati. ALL è il valore predefinito. Per informazioni, consultare Supporto DISTINCT per le aggregazioni bit per bit.

Esempi

È possibile utilizzare le funzioni booleane rispetto alle espressioni booleane o alle espressioni intere. Ad esempio, la seguente query restituisce i risultati dalla tabella USERS standard nel database TICKIT, che ha diverse colonne booleane.

La funzione BOOL_OR restituisce `true` per tutte e cinque le righe. Ad almeno un utente in ciascuno di questi stati piace lo sport.

```
select state, bool_or(likesports) from users
group by state order by state limit 5;

state | bool_or
------+--------
AB    | t
AK    | t
AL    | t
AZ    | t
```

```
BC    | t
(5 rows)
```

Il seguente esempio restituisce NULL.

```
SELECT BOOL_OR(NULL = '123')
               bool_or
------
NULL
```

Espressioni condizionali

Argomenti

- Espressione condizionale CASE
- Funzione DECODE
- Funzioni GREATEST e LEAST
- Funzioni NVL e COALESCE
- Funzione NVL2
- Funzione NULLIF

Amazon Redshift supporta alcune espressioni condizionali che sono estensioni allo standard SQL.

Espressione condizionale CASE

L'espressione CASE è un'espressione condizionale, simile alle istruzioni if (se)/then (quindi)/else (altro) trovate in altre lingue. CASE è utilizzata per specificare un risultato quando ci sono condizioni multiple. Usa CASE quando un'espressione SQL è valida, ad esempio in un comando SELECT.

Esistono due tipi di espressioni CASE: semplici e ricercate.

- Nelle espressioni CASE semplici, un'espressione viene confrontata con un valore. Quando viene trovata una corrispondenza, viene applicata l'azione specificata nella clausola THEN. Se non viene trovata una corrispondenza, viene applicata l'azione nella clausola ELSE.
- Nelle espressioni CASE cercate, ogni CASE viene valutata in base a un'espressione booleana e l'istruzione CASE restituisce la prima CASE corrispondente. Se non vengono trovate corrispondenze tra le clausole WHEN, viene restituita l'operazione nella clausola ELSE.

Sintassi

Semplice istruzione CASE usata per abbinare le condizioni:

```
CASE expression
  WHEN value THEN result
  [WHEN...]
  [ELSE result]
END
```

Istruzione CASE ricercata usata per valutare ogni condizione:

```
CASE
  WHEN condition THEN result
  [WHEN ...]
  [ELSE result]
END
```

Argomenti

espressione

Un nome di colonna o qualsiasi espressione valida.

value

Valore con cui viene confrontata l'espressione, ad esempio una costante numerica o una stringa di caratteri.

result

Il valore o espressione di destinazione che viene restituito quando viene valutata un'espressione o una condizione booleana. I tipi di dati di tutte le espressioni dei risultati devono essere convertibili in un singolo tipo di output.

condizione

Un'espressione booleana che restituisce true o false. Se la condizione è true, il valore dell'espressione CASE è il risultato che segue la condizione e il resto dell'espressione CASE non viene elaborato. Se la condizione è false, vengono valutate tutte le clausole WHEN successive. Se nessun risultato della condizione WHEN è true, il valore dell'espressione CASE è il risultato della clausola ELSE. Se la clausola ELSE viene omessa e nessuna condizione è true, il risultato è null.

Esempi

Gli esempi seguenti utilizzano la tabella VENUE e la tabella SALES dai dati di esempio di TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Utilizzare una semplice espressione CASE per sostituire `New York City` con `Big Apple` in una query sulla tabella VENUE. Sostituire tutti gli altri nomi di città con `other`.

```
select venuecity,
  case venuecity
    when 'New York City'
    then 'Big Apple' else 'other'
  end
from venue
order by venueid desc;

venuecity        |   case
-----------------+-----------
Los Angeles      | other
New York City    | Big Apple
San Francisco    | other
Baltimore        | other
...
```

Utilizzare un'espressione CASE ricercata per assegnare numeri di gruppo in base al valore PRICEPAID per le vendite di biglietti singoli:

```
select pricepaid,
  case when pricepaid <10000 then 'group 1'
    when pricepaid >10000 then 'group 2'
    else 'group 3'
  end
from sales
order by 1 desc;

pricepaid |  case
----------+---------
12624     | group 2
10000     | group 3
10000     | group 3
9996      | group 1
9988      | group 1
```

```
...
```

Funzione DECODE

Un'espressione DECODE sostituisce un valore specifico con un altro valore specifico o un valore predefinito, in base al risultato di una condizione di uguaglianza. Questa operazione è equivalente all'operazione di un'espressione CASE semplice o di un'istruzione IF-THEN-ELSE.

Sintassi

```
DECODE ( expression, search, result [, search, result ]... [ ,default ] )
```

Questo tipo di espressione è utile per sostituire abbreviazioni o codici archiviati in tabelle con valori aziendali significativi necessari per i report.

Parametri

espressione

La fonte del valore che si desidera confrontare, come ad esempio una colonna in una tabella.

cerca

Il valore di destinazione che viene confrontato con l'espressione di origine, ad esempio un valore numerico o una stringa di caratteri. L'espressione di ricerca deve valutare un singolo valore fisso. Non è possibile specificare un'espressione che valuti un intervallo di valori, ad esempio `age between 20 and 29`; è necessario specificare coppie di ricerca/risultato separate per ciascun valore che si desidera sostituire.

Il tipo di dati di tutte le istanze dell'espressione di ricerca deve essere lo stesso o compatibile. I parametri espressione e cerca devono essere anche compatibili.

result

Il valore di sostituzione che la query restituisce quando l'espressione corrisponde al valore di ricerca. È necessario includere almeno una coppia di ricerca/risultato nell'espressione DECODE.

I tipi di dati di tutte le istanze dell'espressione del risultato devono essere gli stessi o compatibili. I parametri risultato e impostazione predefinita devono essere anche compatibili.

default

Un valore predefinito facoltativo che viene utilizzato per i casi in cui la condizione di ricerca non ha esito positivo. Se non viene specificato un parametro, l'espressione DECODE restituisce NULL.

Note per l'utilizzo

Se il valore espressione e il valore cerca sono entrambi NULL, il risultato DECODE è il valore risultato corrispondente. Per un'illustrazione di questo uso della funzione, vedere la sezione Esempi.

Quando viene utilizzato in questo modo, DECODE è simile a Funzione NVL2, ma ci sono alcune differenze. Per una descrizione di queste differenze, vedere le note di utilizzo NVL2.

Esempi

Quando il valore `2008-06-01` esiste nella colonna CALDATE della DATETABLE, l'esempio seguente lo sostituisce con `June 1st, 2008`. L'esempio sostituisce tutti gli altri valori CALDATE con NULL.

```
select decode(caldate, '2008-06-01', 'June 1st, 2008')
from datetable where month='JUN' order by caldate;

case
----------------
June 1st, 2008

...
(30 rows)
```

Nell'esempio seguente viene utilizzata un'espressione DECODE per convertire le cinque colonne CATNAME abbreviate nella tabella CATEGORY in nomi completi e convertire altri valori nella colonna in `Unknown`.

```
select catid, decode(catname,
'NHL', 'National Hockey League',
'MLB', 'Major League Baseball',
'MLS', 'Major League Soccer',
'NFL', 'National Football League',
'NBA', 'National Basketball Association',
'Unknown')
from category
order by catid;

catid  | case
-------+---------------------------------
1      | Major League Baseball
2      | National Hockey League
3      | National Football League
```

```
4      | National Basketball Association
5      | Major League Soccer
6      | Unknown
7      | Unknown
8      | Unknown
9      | Unknown
10     | Unknown
11     | Unknown
(11 rows)
```

Utilizzare un'espressione DECODE per trovare luoghi in Colorado e Nevada con NULL nella colonna VENUESEATS; convertire i valori NULL in zeri. Se la colonna VENUESEATS non è NULL, restituire 1 come risultato.

```
select venuename, venuestate, decode(venueseats,null,0,1)
from venue
where venuestate in('NV','CO')
order by 2,3,1;

venuename                | venuestate     | case
-----------------------------+----------------+-----------
Coors Field                    | CO         |   1
Dick's Sporting Goods Park     | CO         |   1
Ellie Caulkins Opera House     | CO         |   1
INVESCO Field         | CO         |   1
Pepsi Center         | CO         |   1
Ballys Hotel         | NV         |   0
Bellagio Hotel                 | NV         |   0
Caesars Palace                 | NV         |   0
Harrahs Hotel                  | NV         |   0
Hilton Hotel                   | NV         |   0
...
(20 rows)
```

Funzioni GREATEST e LEAST

Restituisce il valore più grande o più piccolo da un elenco numeri di espressioni.

Sintassi

```
GREATEST (value [, ...])
LEAST (value [, ...])
```

Parametri

expression_list

Un elenco di espressioni separate da virgole, come ad esempio i nomi di colonne. Le espressioni devono essere tutte convertibili in un tipo di dati comune. I valori NULL nell'elenco vengono ignorati. Se tutte le espressioni vengono valutate su NULL, il risultato è NULL.

Valori restituiti

Restituisce il valore massimo (per GREATEST) o minimo (per LEAST) dell'elenco di espressioni fornito.

Esempio

Nell'esempio seguente viene restituito il valore più alto in ordine alfabetico per `firstname` oppure `lastname`.

```
select firstname, lastname, greatest(firstname,lastname) from users
where userid < 10
order by 3;

 firstname | lastname  | greatest
-----------+-----------+-----------
 Lars      | Ratliff   | Ratliff
 Reagan    | Hodge     | Reagan
 Colton    | Roy       | Roy
 Barry     | Roy       | Roy
 Tamekah   | Juarez    | Tamekah
 Rafael    | Taylor    | Taylor
 Victor    | Hernandez | Victor
 Vladimir  | Humphrey  | Vladimir
 Mufutau   | Watkins   | Watkins
(9 rows)
```

Funzioni NVL e COALESCE

Restituisce il valore della prima espressione che non è null in una serie di espressioni. Quando viene trovato un valore non null, le restanti espressioni nell'elenco non vengono valutate.

NVL è identica a COALESCE. Sono sinonimi. Questo argomento illustra la sintassi e contiene esempi per entrambe.

Sintassi

```
NVL( expression, expression, ... )
```

La sintassi di COALESCE è la stessa:

```
COALESCE( expression, expression, ... )
```

Se tutte le espressioni sono null, il risultato è null.

Queste funzioni sono utili quando si desidera restituire un valore secondario quando manca un valore primario o è null. Ad esempio, una query potrebbe restituire il primo dei tre numeri di telefono disponibili: cellulare, casa o ufficio. L'ordine delle espressioni nella funzione determina l'ordine di valutazione.

Argomenti

espressione

Un'espressione, come ad esempio un nome di colonna, da valutare per lo stato null.

Tipo restituito

Amazon Redshift determina il tipo di dati del valore restituito in base alle espressioni di input. Se i tipi di dati delle espressioni di input non hanno un tipo comune, viene restituito un errore.

Esempi

Se l'elenco contiene espressioni intere, la funzione restituisce un numero intero.

```
SELECT COALESCE(NULL, 12, NULL);

coalesce
--------------
12
```

Questo esempio, che è uguale all'esempio precedente tranne per il fatto che utilizza NVL, restituisce lo stesso risultato.

```
SELECT NVL(NULL, 12, NULL);
```

```
coalesce
--------------
12
```

Nell'esempio seguente viene restituito un tipo di stringa.

```
SELECT COALESCE(NULL, 'Amazon Redshift', NULL);

coalesce
--------------
Amazon Redshift
```

L'esempio seguente genera un errore perché i tipi di dati variano nell'elenco delle espressioni. In questo caso, nell'elenco sono presenti sia un tipo di stringa che un tipo numerico.

```
SELECT COALESCE(NULL, 'Amazon Redshift', 12);
ERROR: invalid input syntax for integer: "Amazon Redshift"
```

Per questo esempio, si crea una tabella con le colonne START_DATE e END_DATE, si inseriscono alcune righe che includono valori null, quindi si applica un'espressione NVL alle due colonne.

```
create table datetable (start_date date, end_date date);
insert into datetable values ('2008-06-01','2008-12-31');
insert into datetable values (null,'2008-12-31');
insert into datetable values ('2008-12-31',null);
```

```
select nvl(start_date, end_date)
from datetable
order by 1;

coalesce
------------
2008-06-01
2008-12-31
2008-12-31
```

Il nome della colonna predefinita per un'espressione NVL è COALESCE. La seguente query restituisce gli stessi risultati:

```
select coalesce(start_date, end_date)
from datetable
```

```
order by 1;
```

Per le seguenti query di esempio, si crea una tabella con informazioni di esempio sulla prenotazione alberghiera e si inseriscono diverse righe. Alcuni record contengono valori null.

```
create table booking_info (booking_id int, booking_code character(8), check_in date,
 check_out date, funds_collected numeric(12,2));
```

Inserire i seguenti dati di esempio. Alcuni record non hanno una data di `check_out` o un importo `funds_collected`.

```
insert into booking_info values (1, 'OCEAN_WV', '2023-02-01','2023-02-03',100.00);
insert into booking_info values (2, 'OCEAN_WV', '2023-04-22','2023-04-26',120.00);
insert into booking_info values (3, 'DSRT_SUN', '2023-03-13','2023-03-16',125.00);
insert into booking_info values (4, 'DSRT_SUN', '2023-06-01','2023-06-03',140.00);
insert into booking_info values (5, 'DSRT_SUN', '2023-07-10',null,null);
insert into booking_info values (6, 'OCEAN_WV', '2023-08-15',null,null);
```

La seguente query restituisce un elenco di date. Se la data di `check_out` non è disponibile, elenca la data di `check_in`.

```
select coalesce(check_out, check_in)
from booking_info
order by booking_id;
```

Di seguito sono riportati i risultati. Si noti che gli ultimi due record mostrano la data di `check_in`.

```
coalesce
------------
2023-02-03
2023-04-26
2023-03-16
2023-06-03
2023-07-10
2023-08-15
```

Se si prevede che una query restituisca valori null per determinate funzioni o colonne, è possibile utilizzare un'espressione NVL per sostituire i valori null con qualche altro valore. Ad esempio, le funzioni di aggregazione, come SUM, restituiscono valori null anziché zero quando non hanno righe da valutare. È possibile utilizzare un'espressione NVL per sostituire questi valori null con `700.0`.

Invece di 485, il risultato della somma di funds_collected è 1885 perché due righe con valori null vengono sostituite con 700.

```
select sum(nvl(funds_collected, 700.0)) as sumresult from booking_info;

sumresult
------
 1885
```

Funzione NVL2

Restituisce uno dei due valori in base al fatto che un'espressione specificata valuti in NULL o NOT NULL.

Sintassi

```
NVL2 ( expression, not_null_return_value, null_return_value )
```

Argomenti

espressione

Un'espressione, come ad esempio un nome di colonna, da valutare per lo stato null.

not_null_return_value

Il valore restituito se l' espressione valuta in NOT NULL. Il valore not_null_return_value deve avere lo stesso tipo di dati dell' espressione o essere implicitamente convertibile in quel tipo di dati.

null_return_value

Il valore restituito se l' espressione valuta in NULL. Il valore null_return_value deve avere lo stesso tipo di dati dell' espressione o essere implicitamente convertibile in quel tipo di dati.

Tipo restituito

Il tipo di restituzione NVL2 è determinato nel modo seguente:

- Se not_null_return_value o null_return_value è null, viene restituito il tipo di dati dell'espressione not null.

Se sia not_null_return_value sia null_return_value sono non null:

- Se sia not_null_return_value sia null_return_value hanno lo stesso tipo di dati, viene restituito quel tipo di dati.
- Se not_null_return_value e null_return_value hanno tipi di dati numerici diversi, viene restituito il tipo di dati numerico compatibile più piccolo.
- Se not_null_return_value e null_return_value hanno tipi di dati di datetime diversi, viene restituito un tipo di dati numerici di timestamp.
- Se not_null_return_value e null_return_value hanno diversi tipi di dati di carattere, viene restituito il tipo di dati di not_null_return_value.
- Se not_null_return_value e null_return_value hanno tipi di dati numerici e non numerici misti viene restituito il tipo di dati di not_null_return_value.

> **Important**
>
> Negli ultimi due casi in cui viene restituito il tipo di dati di not_null_return_value, null_return_value viene assegnato implicitamente a quel tipo di dati. Se i tipi di dati non sono compatibili, la funzione ha esito negativo.

Note per l'utilizzo

Funzione DECODE può essere utilizzato in modo simile a NVL2 quando i parametri espressione e cerca sono entrambi null. La differenza è che per DECODE, la restituzione avrà sia il valore sia il tipo di dati del parametro risultato. Al contrario, per NVL2, la restituzione avrà il valore di entrambi i parametri, not_null_return_value oppure null_return_value, qualunque sia selezionato dalla funzione, ma avrà il tipo di dati di not_null_return_value.

Ad esempio, supponendo che la colonna 1 sia NULL, le seguenti query restituiranno lo stesso valore. Tuttavia, il tipo di dati del valore di restituzione DECODE sarà INTEGER e il tipo di dati del valore di restituzione NVL2 sarà VARCHAR.

```
select decode(column1, null, 1234, '2345');
select nvl2(column1, '2345', 1234);
```

Esempio

L'esempio seguente modifica alcuni dati di esempio, quindi valuta due campi per fornire informazioni di contatto appropriate per gli utenti:

```
update users set email = null where firstname = 'Aphrodite' and lastname = 'Acevedo';

select (firstname + ' ' + lastname) as name,
nvl2(email, email, phone) AS contact_info
from users
where state = 'WA'
and lastname  like 'A%'
order by lastname, firstname;

name        contact_info
--------------------+--------------------------------------------
Aphrodite Acevedo (906) 632-4407
Caldwell Acevedo  Nunc.sollicitudin@Duisac.ca
Quinn Adams    vel@adipiscingligulaAenean.com
Kamal Aguilar  quis@vulputaterisusa.com
Samson Alexander  hendrerit.neque@indolorFusce.ca
Hall Alford    ac.mattis@vitaediamProin.edu
Lane Allen     et.netus@risusDonec.org
Xander Allison    ac.facilisis.facilisis@Infaucibus.com
Amaya Alvarado    dui.nec.tempus@eudui.edu
Vera Alvarez   at.arcu.Vestibulum@pellentesque.edu
Yetta Anthony  enim.sit@risus.org
Violet Arnold  ad.litora@at.com
August Ashley  consectetuer.euismod@Phasellus.com
Karyn Austin   ipsum.primis.in@Maurisblanditenim.org
Lucas Ayers    at@elitpretiumet.com
```

Funzione NULLIF

Sintassi

L'espressione NULLIF confronta due argomenti e restituisce null se gli argomenti sono uguali. Se non sono uguali, viene restituito il primo argomento. Questa espressione è l'inverso dell'espressione NVL o COALESCE.

```
NULLIF ( expression1, expression2 )
```

Argomenti

expression1, expression2

Le colonne o le espressioni di destinazione che vengono confrontate. Il tipo di restituzione è uguale al tipo della prima espressione. Il nome della colonna predefinito del risultato NULLIF è il nome della colonna della prima espressione.

Esempi

Nell'esempio seguente, la query restituisce la stringa `first` perché gli argomenti non sono uguali.

```
SELECT NULLIF('first', 'second');

case
-------
first
```

Nell'esempio seguente, la query restituisce NULL perché gli argomenti della stringa letterale non sono uguali.

```
SELECT NULLIF('first', 'first');

case
-------
NULL
```

Nell'esempio seguente, la query restituisce 1 perché gli argomenti del numero intero non sono uguali.

```
SELECT NULLIF(1, 2);

case
-------
1
```

Nell'esempio seguente, la query restituisce NULL perché gli argomenti del numero intero sono uguali.

```
SELECT NULLIF(1, 1);
```

```
case
-------
NULL
```

Nell'esempio seguente, la query restituisce null quando i valori LISTID e SALESID corrispondono:

```
select nullif(listid,salesid), salesid
from sales where salesid<10 order by 1, 2 desc;

listid  | salesid
--------+---------
     4  |       2
     5  |       4
     5  |       3
     6  |       5
     10 |       9
     10 |       8
     10 |       7
     10 |       6
        |       1
(9 rows)
```

È possibile utilizzare NULLIF per garantire che le stringhe vuote vengano sempre restituite come null. Nell'esempio seguente, l'espressione NULLIF restituisce un valore null o una stringa che contiene almeno un carattere.

```
insert into category
values(0,'','Special','Special');

select nullif(catgroup,'') from category
where catdesc='Special';

catgroup
----------
null
(1 row)
```

NULLIF ignora gli spazi finali. Se una stringa non è vuota ma contiene spazi, NULLIF restituisce ancora null:

```
create table nulliftest(c1 char(2), c2 char(2));
```

```
insert into nulliftest values ('a','a ');

insert into nulliftest values ('b','b');


select nullif(c1,c2) from nulliftest;
c1
------
null
null
(2 rows)
```

Funzioni di formattazione del tipo di dati

Argomenti

- Funzione CAST
- Funzione CONVERT
- TO_CHAR
- Funzione TO_DATE
- TO_NUMBER
- TEXT_TO_INT_ALT
- TEXT_TO_NUMERIC_ALT
- Stringhe di formato datetime
- Stringhe di formato numerico
- Caratteri di formattazione in stile Teradata per i dati numerici

Le funzioni di formattazione del tipo di dati forniscono un modo semplice per convertire i valori da un tipo di dati a un altro. Per ognuna di queste funzioni, il primo argomento è sempre il valore da formattare e il secondo argomento contiene il modello per il nuovo formato. Amazon Redshift supporta diverse funzioni di formattazione del tipo di dati.

Funzione CAST

La funzione CAST converte un tipo di dati in un altro tipo di dati compatibile. Ad esempio, puoi convertire una stringa in una data o un tipo numerico in una stringa. CAST esegue una conversione

in fase di runtime, il che significa che la conversione non modifica il tipo di dati di un valore in una tabella di origine. Viene modificato solo nel contesto della query.

La funzione CAST è molto simile a the section called “CONVERT”, in quanto entrambe eseguono la conversione da un tipo di dati all'altro, ma vengono chiamate in modo diverso.

Alcuni tipi di dati richiedono una conversione esplicita in altri tipi di dati utilizzando la funzione CAST o CONVERT. Altri tipi di dati possono essere convertiti implicitamente, come parte di un altro comando, senza utilizzare CAST o CONVERT. Per informazioni, consulta Conversione e compatibilità dei tipi.

Sintassi

Utilizza i seguenti due formati di sintassi equivalenti per convertire espressioni da un tipo di dati a un altro:

```
CAST ( expression AS type )
expression :: type
```

Argomenti

espressione

Un'espressione che valuta uno o più valori, ad esempio un nome di colonna o un letterale. La conversione di valori null restituisce null. L'espressione non può contenere stringhe o spazi vuoti.

tipo

Uno dei metodi supportati Tipi di dati.

Tipo restituito

CAST restituisce il tipo di dati specificato dall'argomento tipo.

> **Note**
>
> Amazon Redshift restituisce un errore se provi a eseguire una conversione problematica, come la seguente conversione DECIMAL che perde precisione:
>
> ```
> select 123.456::decimal(2,1);
> ```
>
> o una conversione INTEGER che provoca un eccesso:

```
select 12345678::smallint;
```

Esempi

La maggior parte degli esempi usa il database TICKIT di esempio. Per ulteriori informazioni sull'impostazione dei dati di esempio, consulta Load data.

Le seguenti due query sono equivalenti. Entrambi assegnano un valore decimale a un integer:

```
select cast(pricepaid as integer)
from sales where salesid=100;

pricepaid
-----------
162
(1 row)
```

```
select pricepaid::integer
from sales where salesid=100;

pricepaid
-----------
162
(1 row)
```

La seguente query produce un risultato simile. Non richiede dati di esempio per l'esecuzione:

```
select cast(162.00 as integer) as pricepaid;

pricepaid
-----------
162
(1 row)
```

In questo esempio, i valori in una colonna timestamp vengono convertiti come date, il che comporta la rimozione dell'ora da ogni risultato:

```
select cast(saletime as date), salesid
```

```
from sales order by salesid limit 10;

 saletime  | salesid
-----------+---------
2008-02-18 |       1
2008-06-06 |       2
2008-06-06 |       3
2008-06-09 |       4
2008-08-31 |       5
2008-07-16 |       6
2008-06-26 |       7
2008-07-10 |       8
2008-07-22 |       9
2008-08-06 |      10
(10 rows)
```

Se non hai usato CAST come illustrato nell'esempio precedente, i risultati includono l'ora: 18-02-2008 02:36:48.

La seguente query converte i dati di caratteri variabili in una data. Non richiede dati di esempio per l'esecuzione.

```
select cast('2008-02-18 02:36:48' as date) as mysaletime;

mysaletime
--------------------
2008-02-18
(1 row)
```

In questo esempio, il casting dei valori in una colonna di data viene eseguito come timestamps:

```
select cast(caldate as timestamp), dateid
from date order by dateid limit 10;

      caldate       | dateid
--------------------+--------
2008-01-01 00:00:00 |   1827
2008-01-02 00:00:00 |   1828
2008-01-03 00:00:00 |   1829
2008-01-04 00:00:00 |   1830
2008-01-05 00:00:00 |   1831
2008-01-06 00:00:00 |   1832
```

```
2008-01-07 00:00:00 |   1833
2008-01-08 00:00:00 |   1834
2008-01-09 00:00:00 |   1835
2008-01-10 00:00:00 |   1836
(10 rows)
```

In un caso come nell'esempio precedente, è possibile ottenere un ulteriore controllo sulla formattazione dell'output utilizzando TO_CHAR.

In questo esempio, un integer è assegnato come una stringa di caratteri:

```
select cast(2008 as char(4));

bpchar
--------
2008
```

In questo esempio, un valore DECIMAL(6,3) viene assegnato come valore DECIMAL(4,1):

```
select cast(109.652 as decimal(4,1));

numeric
---------
109.7
```

Questo esempio mostra un'espressione più complessa. La colonna PRICEPAID (una colonna DECIMAL(8,2)) nella tabella SALES viene convertita in una colonna DECIMAL(38,2) e i valori vengono moltiplicati per 100000000000000000000:

```
select salesid, pricepaid::decimal(38,2)*100000000000000000000
as value from sales where salesid<10 order by salesid;


 salesid |           value
---------+----------------------------
       1 | 72800000000000000000000.00
       2 |  76000000000000000000000.00
       3 | 35000000000000000000000.00
       4 | 17500000000000000000000.00
       5 | 15400000000000000000000.00
       6 | 39400000000000000000000.00
       7 | 78800000000000000000000.00
```

```
       8 | 197000000000000000000000.00
       9 | 591000000000000000000000.00
(9 rows)
```

> **Note**
>
> Non è possibile eseguire un'operazione CAST o CONVERT sul tipo di dati GEOMETRY per modificarlo in un altro tipo di dati. Tuttavia, è possibile fornire una rappresentazione esadecimale di una stringa letterale in formato Extended Well-Known Binary (EWKB) come input per funzioni che accettano un argomento di tipo GEOMETRY. Ad esempio, la seguente funzione ST_AsText si aspetta un tipo di dati GEOMETRY.
>
> ```
> SELECT ST_AsText('010100000000000000000001C40000000000002040');
> ```
>
> ```
> st_astext
> ------------
>  POINT(7 8)
> ```
>
> È anche possibile specificare esplicitamente il tipo di dati GEOMETRY.
>
> ```
> SELECT ST_AsText('010100000000000000000014400000000000001840'::geometry);
> ```
>
> ```
> st_astext
> ------------
>  POINT(5 6)
> ```

Funzione CONVERT

Come la funzione CAST, la funzione CONVERT converte un tipo di dati in un altro tipo di dati compatibile. Ad esempio, puoi convertire una stringa in una data o un tipo numerico in una stringa. CONVERT esegue una conversione in fase di runtime, il che significa che la conversione non modifica il tipo di dati di un valore in una tabella di origine. Viene modificato solo nel contesto della query.

Alcuni tipi di dati richiedono una conversione esplicita in altri tipi di dati utilizzando la funzione CONVERT. Altri tipi di dati possono essere convertiti implicitamente, come parte di un altro comando, senza utilizzare CAST o CONVERT. Per informazioni, consulta Conversione e compatibilità dei tipi.

Sintassi

```
CONVERT ( type, expression )
```

Argomenti

tipo

Uno dei metodi supportati Tipi di dati.

espressione

Un'espressione che valuta uno o più valori, ad esempio un nome di colonna o un letterale. La conversione di valori null restituisce null. L'espressione non può contenere stringhe o spazi vuoti.

Tipo restituito

CONVERT restituisce il tipo di dati specificato dall'argomento tipo.

> **Note**
>
> Amazon Redshift restituisce un errore se provi a eseguire una conversione problematica, come la seguente conversione DECIMAL che perde precisione:
>
> ```
> SELECT CONVERT(decimal(2,1), 123.456);
> ```
>
> o una conversione INTEGER che provoca un eccesso:
>
> ```
> SELECT CONVERT(smallint, 12345678);
> ```

Esempi

La maggior parte degli esempi usa il database TICKIT di esempio. Per ulteriori informazioni sull'impostazione dei dati di esempio, consulta Caricare dati.

La seguente query utilizza la funzione CONVERT per convertire una colonna di decimali in numeri interi

```
SELECT CONVERT(integer, pricepaid)
```

```
FROM sales WHERE salesid=100;
```

Questo esempio converte un numero intero in una stringa di caratteri.

```
SELECT CONVERT(char(4), 2008);
```

In questo esempio, la data e l'ora correnti vengono convertite in un tipo di dati di caratteri variabili:

```
SELECT CONVERT(VARCHAR(30), GETDATE());

getdate
---------
2023-02-02 04:31:16
```

Questo esempio converte la colonna saletime nella sola ora, rimuovendo le date da ogni riga.

```
SELECT CONVERT(time, saletime), salesid
FROM sales order by salesid limit 10;
```

Per informazioni sulla conversione di un timestamp da un fuso orario a un altro, consulta Funzione CONVERT_TIMEZONE. Per ulteriori funzioni di data e ora, consulta Funzioni di data e ora.

L'esempio seguente converte i dati di caratteri variabili in un oggetto datetime.

```
SELECT CONVERT(datetime, '2008-02-18 02:36:48') as mysaletime;
```

> **Note**
>
> Non è possibile eseguire un'operazione CAST o CONVERT sul tipo di dati `GEOMETRY` per modificarlo in un altro tipo di dati. Tuttavia, è possibile fornire una rappresentazione esadecimale di una stringa letterale in formato Extended Well-Known Binary (EWKB) come input per funzioni che accettano un argomento di tipo `GEOMETRY`. Ad esempio, la seguente funzione `ST_AsText` si aspetta un tipo di dati `GEOMETRY`.
>
> ```
> SELECT ST_AsText('010100000000000000000001C400000000000002040');
> ```
>
> ```
> st_astext
> ------------
>  POINT(7 8)
> ```

È anche possibile specificare esplicitamente il tipo di dati GEOMETRY.

```
SELECT ST_AsText('0101000000000000000000144000000000000001840'::geometry);
```

```
st_astext
------------
 POINT(5 6)
```

TO_CHAR

TO_CHAR converte un timestamp o un'espressione numerica in un formato di dati carattere-stringa.

Sintassi

```
TO_CHAR (timestamp_expression | numeric_expression , 'format')
```

Argomenti

timestamp_expression

Un'espressione che restituisce un valore di tipo TIMESTAMP o TIMESTAMPTZ o un valore che può essere implicitamente costretto in un timestamp.

numeric_expression

Un'espressione che restituisce un valore di tipo di dati numerici un valore che può essere implicitamente costretto in un tipo numerico. Per ulteriori informazioni, consultare Tipi numerici. TO_CHAR inserisce uno spazio a sinistra della stringa numerica.

> **Note**
>
> TO_CHAR non supporta i valori DECIMAL a 128 bit.

format

Il formato per il nuovo valore. Per i formati validi, consultare Stringhe di formato datetime e Stringhe di formato numerico.

Tipo restituito

VARCHAR

Esempi

Nell'esempio seguente un timestamp viene convertito in un valore con la data e l'ora in un formato con il nome del mese a nove caratteri, il nome del giorno della settimana e il numero del giorno del mese.

```
select to_char(timestamp '2009-12-31 23:15:59', 'MONTH-DY-DD-YYYY HH12:MIPM');

to_char
-------------------------
DECEMBER -THU-31-2009 11:15PM
```

Nell'esempio seguente un timestamp viene convertito in un valore con il numero di giorno dell'anno.

```
select to_char(timestamp '2009-12-31 23:15:59', 'DDD');

to_char
-------------------------
365
```

Nell'esempio seguente un timestamp viene convertito in un numero di giorno ISO della settimana.

```
select to_char(timestamp '2022-05-16 23:15:59', 'ID');

to_char
-------------------------
1
```

L'esempio seguente estrae il nome del mese da un valore di data.

```
select to_char(date '2009-12-31', 'MONTH');

to_char
-------------------------
DECEMBER
```

Nell'esempio seguente viene convertito ogni valore STARTTIME nella tabella EVENT in una stringa composta da ore, minuti e secondi.

```
select to_char(starttime, 'HH12:MI:SS')
from event where eventid between 1 and 5
order by eventid;

to_char
----------
02:30:00
08:00:00
02:30:00
02:30:00
07:00:00
```

L'esempio seguente converte un intero valore timestamp in un formato diverso.

```
select starttime, to_char(starttime, 'MON-DD-YYYY HH12:MIPM')
from event where eventid=1;

      starttime      |       to_char
---------------------+---------------------
 2008-01-25 14:30:00 | JAN-25-2008 02:30PM
```

L'esempio seguente converte un letterale di timestamp in una stringa di caratteri.

```
select to_char(timestamp '2009-12-31 23:15:59','HH24:MI:SS');

to_char
----------
23:15:59
```

L'esempio seguente converte un numero decimale in una stringa di caratteri.

```
select to_char(125.8, '999.99');

to_char
---------
125.80
```

L'esempio seguente converte un numero decimale in una stringa di caratteri.

```
select to_char(125.8, '999D99');
```

```
to_char
---------
125.80
```

L'esempio seguente converte un numero in una stringa di caratteri con uno zero iniziale.

```
select to_char(125.8, '0999D99');

to_char
---------
0125.80
```

L'esempio seguente converte un numero in una stringa di caratteri con il segno negativo alla fine.

```
select to_char(-125.8, '999D99S');

to_char
---------
125.80-
```

L'esempio seguente converte un numero in una stringa di caratteri con il segno positivo o negativo nella posizione specificata.

```
select to_char(125.8, '999D99SG');

to_char
---------
125.80+
```

L'esempio seguente converte un numero in una stringa di caratteri con il segno positivo nella posizione specificata.

```
select to_char(125.8, 'PL999D99');

to_char
---------
+ 125.80
```

L'esempio seguente converte un numero in una stringa di caratteri con il simbolo di valuta.

```
select to_char(-125.88, '$S999D99');

to_char
---------
$-125.88
```

L'esempio seguente converte un numero in una stringa di caratteri con il simbolo di valuta nella posizione specificata.

```
select to_char(-125.88, 'S999D99L');

to_char
---------
-125.88$
```

L'esempio seguente converte un numero in una stringa di caratteri utilizzando un separatore di migliaia (virgole).

```
select to_char(1125.8, '9,999.99');

to_char
---------
1,125.80
```

L'esempio seguente converte un numero in una stringa di caratteri con parentesi angolari per i numeri negativi.

```
select to_char(-125.88, '$999D99PR');

to_char
---------
$<125.88>
```

L'esempio seguente converte un numero in una stringa di numeri romani.

```
select to_char(125, 'RN');

to_char
---------
   CXXV
```

L'esempio seguente converte una data in un codice centenario.

```
select to_char(date '2020-12-31', 'CC');

to_char
---------
21
```

L'esempio seguente mostra il giorno della settimana.

```
SELECT to_char(current_timestamp, 'FMDay, FMDD HH12:MI:SS');

to_char
-----------------------
Wednesday, 31 09:34:26
```

L'esempio seguente visualizza il suffisso numerico ordinale per un numero.

```
SELECT to_char(482, '999th');

to_char
-----------------------
 482nd
```

L'esempio seguente sottrae la commissione dal prezzo pagato nella tabella delle vendite. La differenza viene quindi arrotondata e convertita in un numero romano, mostrato nella colonna to_char:

```
select salesid, pricepaid, commission, (pricepaid - commission)
as difference, to_char(pricepaid - commission, 'rn') from sales
group by sales.pricepaid, sales.commission, salesid
order by salesid limit 10;

 salesid | pricepaid | commission | difference |     to_char
---------+-----------+------------+------------+-----------------
       1 |    728.00 |     109.20 |     618.80 |           dcxix
       2 |     76.00 |      11.40 |      64.60 |             lxv
       3 |    350.00 |      52.50 |     297.50 |        ccxcviii
       4 |    175.00 |      26.25 |     148.75 |           cxlix
       5 |    154.00 |      23.10 |     130.90 |           cxxxi
       6 |    394.00 |      59.10 |     334.90 |         cccxxxv
       7 |    788.00 |     118.20 |     669.80 |           dclxx
```

```
       8 |    197.00 |      29.55 |     167.45 |          clxvii
       9 |    591.00 |      88.65 |     502.35 |             dii
      10 |     65.00 |       9.75 |      55.25 |              lv
```

L'esempio seguente aggiunge il simbolo di valuta ai valori di differenza mostrati nella colonna to_char:

```
select salesid, pricepaid, commission, (pricepaid - commission)
as difference, to_char(pricepaid - commission, 'l99999D99') from sales
group by sales.pricepaid, sales.commission, salesid
order by salesid limit 10;

salesid | pricepaid | commission | difference |  to_char
--------+-----------+------------+------------+------------
      1 |    728.00 |     109.20 |     618.80 | $   618.80
      2 |     76.00 |      11.40 |      64.60 | $    64.60
      3 |    350.00 |      52.50 |     297.50 | $   297.50
      4 |    175.00 |      26.25 |     148.75 | $   148.75
      5 |    154.00 |      23.10 |     130.90 | $   130.90
      6 |    394.00 |      59.10 |     334.90 | $   334.90
      7 |    788.00 |     118.20 |     669.80 | $   669.80
      8 |    197.00 |      29.55 |     167.45 | $   167.45
      9 |    591.00 |      88.65 |     502.35 | $   502.35
     10 |     65.00 |       9.75 |      55.25 | $    55.25
```

L'esempio seguente elenca il secolo in cui è stata effettuata ciascuna vendita.

```
select salesid, saletime, to_char(saletime, 'cc') from sales
order by salesid limit 10;

 salesid |      saletime       | to_char
---------+---------------------+---------
       1 | 2008-02-18 02:36:48 | 21
       2 | 2008-06-06 05:00:16 | 21
       3 | 2008-06-06 08:26:17 | 21
       4 | 2008-06-09 08:38:52 | 21
       5 | 2008-08-31 09:17:02 | 21
       6 | 2008-07-16 11:59:24 | 21
       7 | 2008-06-26 12:56:06 | 21
       8 | 2008-07-10 02:12:36 | 21
       9 | 2008-07-22 02:23:17 | 21
      10 | 2008-08-06 02:51:55 | 21
```

Nell'esempio seguente viene convertito ogni valore STARTTIME nella tabella EVENT in una stringa composta da ore, minuti, secondi e fuso orario.

```
select to_char(starttime, 'HH12:MI:SS TZ')
from event where eventid between 1 and 5
order by eventid;

to_char
----------
02:30:00 UTC
08:00:00 UTC
02:30:00 UTC
02:30:00 UTC
07:00:00 UTC
```

L'esempio seguente mostra la formattazione per secondi, millisecondi e microsecondi.

```
select sysdate,
to_char(sysdate, 'HH24:MI:SS') as seconds,
to_char(sysdate, 'HH24:MI:SS.MS') as milliseconds,
to_char(sysdate, 'HH24:MI:SS:US') as microseconds;

timestamp           | seconds  | milliseconds | microseconds
--------------------+----------+--------------+----------------
2015-04-10 18:45:09 | 18:45:09 | 18:45:09.325 | 18:45:09:325143
```

Funzione TO_DATE

TO_DATE converte una data rappresentata con una stringa di caratteri in un tipo di dati DATE.

Sintassi

```
TO_DATE(string, format)
```

```
TO_DATE(string, format, is_strict)
```

Argomenti

stringa

Una stringa da convertire.

format

Una letterale di stringa che definisce il formato della stringa di input,in termini di parti della data. Per un elenco di formati validi per giorno, mese e anno, consultare Stringhe di formato datetime.

is_strict

Un valore booleano facoltativo che specifica se viene restituito un errore se un valore date di input non è compreso nell'intervallo. Quando is_strict è impostato su TRUE, viene restituito un errore se esiste un valore fuori intervallo. Quando is_strict è impostato su FALSE, che è il valore di default, allora i valori di overflow sono accettati.

Tipo restituito

TO_DATE restituisce una DATA, in base al valore formato.

Se la conversione in formato non riesce, viene restituito un errore.

Esempi

L'istruzione SQL seguente converte la data `02 Oct 2001` in un tipo di dati data.

```
select to_date('02 Oct 2001', 'DD Mon YYYY');

to_date
------------
2001-10-02
(1 row)
```

L'istruzione SQL seguente converte la stringa `20010631` in una data.

```
select to_date('20010631', 'YYYYMMDD', FALSE);
```

Il risultato è il 1° luglio 2001, perché a giugno ci sono solo 30 giorni.

```
to_date
------------
2001-07-01
```

L'istruzione SQL seguente converte la stringa `20010631` in una data:

```
to_date('20010631', 'YYYYMMDD', TRUE);
```

Il risultato è un errore perché ci sono solo 30 giorni a giugno.

```
ERROR:  date/time field date value out of range: 2001-6-31
```

TO_NUMBER

TO_NUMBER converte una stringa in un valore numerico (decimale).

Sintassi

```
to_number(string, format)
```

Argomenti

stringa

Stringa da convertire. Il formato deve essere un valore letterale.

format

Il secondo argomento è una stringa di formato che indica come deve essere analizzata la stringa di caratteri per creare il valore numerico. Ad esempio, il formato `'99D999'` specifica che la stringa da convertire è composta da cinque cifre con il punto decimale nella terza posizione. Ad esempio, `to_number('12.345','99D999')` restituisce `12.345` come valore numerico. Per un elenco di formati validi, consultare Stringhe di formato numerico.

Tipo restituito

TO_NUMBER restituisce un numero DECIMAL.

Se la conversione in formato non riesce, viene restituito un errore.

Esempi

L'esempio seguente converte la stringa `12,454.8-` in un numero:

```
select to_number('12,454.8-', '99G999D9S');

to_number
```

```
-----------
-12454.8
```

L'esempio seguente converte la stringa `$ 12,454.88` in un numero:

```
select to_number('$ 12,454.88', 'L 99G999D99');

to_number
-----------
12454.88
```

L'esempio seguente converte la stringa `$ 2,012,454.88` in un numero:

```
select to_number('$ 2,012,454.88', 'L 9,999,999.99');

to_number
-----------
2012454.88
```

TEXT_TO_INT_ALT

TEXT_TO_INT_ALT converte una stringa di caratteri in un numero intero utilizzando la formattazione in stile Teradata. Le cifre di frazione nel risultato vengono troncate.

Sintassi

```
TEXT_TO_INT_ALT (expression [ , 'format'])
```

Argomenti

espressione

Un'espressione che restituisce uno o più valori CHAR o VARCHAR, ad esempio un nome di colonna o una stringa letterale. La conversione di valori null restituisce null. La funzione converte stringhe le vuote in 0.

format

Una letterale di stringa che definisce il formato dell'espressione di input. Per ulteriori informazioni sui caratteri di formattazione che è possibile specificare, consultare Caratteri di formattazione in stile Teradata per i dati numerici.

Tipo restituito

TEXT_TO_INT_ALT restituisce un valore INTEGER.

La parte frazionale del risultato del casting viene troncata.

Amazon Redshift restituisce un errore se la conversione nella frase format che si specifica non ha esito positivo.

Esempi

Nell'esempio seguente la stringa expression di input '123-' viene convertita nel numero intero -123.

```
select text_to_int_alt('123-');
```

```
text_to_int_alt
----------
      -123
```

L'esempio seguente converte la stringa expression di input '2147483647+' nel numero intero 2147483647.

```
select text_to_int_alt('2147483647+');
```

```
text_to_int_alt
----------
2147483647
```

Nell'esempio seguente la stringa expression di input '123E-2' viene convertita nel numero intero -1.

```
select text_to_int_alt('-123E-2');
```

```
text_to_int_alt
----------
        -1
```

L'esempio seguente converte la stringa expression di input '2147483647+' nel numero intero 2147483647.

```
select text_to_int_alt('2147483647+');
```

```
text_to_int_alt
----------
2147483647
```

Nell'esempio seguente la stringa expression di input '123{' con la frase format '999S' viene convertita nel numero intero 1230. Il carattere S indica un decimale con zone firmate. Per ulteriori informazioni, consultare Caratteri di formattazione in stile Teradata per i dati numerici.

```
text_to_int_alt('123{', '999S');
```

```
text_to_int_alt
----------
      1230
```

Nell'esempio seguente la stringa expression di input 'USD123' con la frase format 'C9(I)' viene convertita nel numero intero 123. Per informazioni, consultare Caratteri di formattazione in stile Teradata per i dati numerici.

```
text_to_int_alt('USD123', 'C9(I)');
```

```
text_to_int_alt
----------
       123
```

Nell'esempio seguente viene specificata una colonna di tabella come expression di input.

```
select text_to_int_alt(a), text_to_int_alt(b) from t_text2int order by 1;
```

```
 text_to_int_alt | text_to_int_alt
-----------------+-----------------
            -123 |            -123
            -123 |            -123
             123 |             123
             123 |             123
```

Di seguito è riportata la definizione di tabella e l'istruzione insert per questo esempio.

```
create table t_text2int (a varchar(200), b char(200));
```

```
insert into t_text2int VALUES('123', '123'),('123.123', '123.123'), ('-123', '-123'),
 ('123-', '123-');
```

TEXT_TO_NUMERIC_ALT

TEXT_TO_NUMERIC_ALT esegue un'operazione di casting in stile Teradata per convertire una stringa di caratteri in un formato di dati numerico.

Sintassi

```
TEXT_TO_NUMERIC_ALT (expression [, 'format'] [, precision, scale])
```

Argomenti

espressione

Un'espressione che restituisce uno o più valori CHAR o VARCHAR, ad esempio un nome di colonna o un letterale. La conversione di valori null restituisce null. Le stringhe vuote sono convertite in 0.

format

Una letterale di stringa che definisce il formato dell'espressione di input. Per ulteriori informazioni, consultare Caratteri di formattazione in stile Teradata per i dati numerici.

precisione

Il numero di cifre nel risultato numerico. L'impostazione predefinita è 38.

scale

Il numero di cifre alla destra del punto decimale nel risultato numerico. Il valore predefinito è 0.

Tipo restituito

TEX_TO_NUMERIC_ALT restituisce un numero DECIMAL.

Amazon Redshift restituisce un errore se la conversione nella frase format che si specifica non ha esito positivo.

Amazon Redshift esegue il casting della stringa expression di input sul tipo numerico con la massima precisione specificata per quel tipo nell'opzione precision. Se la lunghezza del valore numerico supera il valore specificato per precision, Amazon Redshift arrotonda il valore numerico in base alle seguenti regole:

- Se la lunghezza del risultato del casting supera la lunghezza specificata nella frase format, Amazon Redshift restituisce un errore.
- Se il risultato viene sottoposto a casting su un valore numerico, viene arrotondato al valore più vicino. Se la porzione frazionaria è esattamente a metà strada tra il risultato del casting superiore e quello inferiore, il risultato viene arrotondato al valore pari più vicino.

Esempi

L'esempio seguente converte la stringa expression di input '1.5' nel valore numerico '2'. Perché l'istruzione non specifica scale, scale è impostato su 0 e il risultato del casting non include un risultato di frazione. Poiché .5 è a metà strada tra 1 e 2, il risultato del casting viene arrotondato al valore pari di 2.

```
select text_to_numeric_alt('1.5');
```

```
 text_to_numeric_alt
---------------------
                   2
```

L'esempio seguente converte la stringa expression di input '2.51' nel valore numerico '3'. Perché l'istruzione non specifica un valore scale, scale è impostato su 0 e il risultato del casting non include un risultato di frazione. Poiché .51 è più vicino a 3 che a 2, il risultato del casting viene arrotondato al valore 3.

```
select text_to_numeric_alt('2.51');
```

```
 text_to_numeric_alt
---------------------
                   3
```

L'esempio seguente converte la stringa expression di input 123.52501 con una precisione di 10 e una scala di 2 al valore numerico 123,53.

```
select text_to_numeric_alt('123.52501', 10, 2);
```

```
 text_to_numeric_alt
---------------------
              123.53
```

Nell'esempio seguente viene specificata una colonna di tabella come expression di input.

```
select text_to_numeric_alt(a), text_to_numeric_alt(b) from t_text2numeric order by 1;
```

```
          text_to_numeric_alt           |           text_to_numeric_alt
----------------------------------------+----------------------------------------
 -99999999999999999999999999999999999999 | -99999999999999999999999999999999999999
                                  -12300 |                                  -12300
                                     123 |                                     123
                                     123 |                                     123
  99999999999999999999999999999999999999 |  99999999999999999999999999999999999999
```

Di seguito è riportata la definizione di tabella e l'istruzione insert per questo esempio.

```
create table  t_text2numeric (a varchar(200), b char(200));
```

```
insert into  t_text2numeric values
('123', '123'),
('+123.456', '+123.456'),
('-' || repeat('9', 38), '-' || repeat('9', 38)),
(repeat('9', 38) || '+', repeat('9', 38) || '+'),
('-123E2', '-123E2');
```

Stringhe di formato datetime

È possibile trovare un riferimento per le stringhe di formato datetime di seguito.

Le seguenti stringhe di formato si applicano a funzioni quali TO_CHAR. Queste stringhe possono contenere separatori di data e ora (come '-', '/' o ':') e i seguenti "dateparts" e "timeparts".

Parte di data o parte di ora	Significato
BC o B.C., AD o A.D., b.c. o bc, ad o a.d.	Indicatori di epoca in maiuscolo o minuscolo
CC	Numero di secolo a due cifre
AAAA, AAA, AA, A	Numero dell'anno di 4 cifre, 3 cifre, 2 cifre, 1 cifra
A, AAA	Numero dell'anno di 4 cifre con una virgola
IAAA, IAA, IA, I	Numero dell'anno dell'Organizzazione internazionale per la standardizzazione (ISO) di 4 cifre, 3 cifre, 2 cifre, 1 cifra
Q	Numero del trimestre (da 1 a 4).
MESE, Mese, mese	Nome del mese (maiuscolo, maiuscolo e minuscolo, minuscolo, riempito a vuoto con 9 caratteri)
MES, Mes, mes	Nome del mese abbreviato (maiuscolo, maiuscolo e minuscolo, minuscolo, riempito a vuoto con 3 caratteri)
MM	Numero del mese (01-12)
RM, rm	Il numero del mese in numeri romani (I-XII, con I che corrisponde a gennaio, maiuscolo o minuscolo)
W	La settimana del mese (1-5, la prima settimana inizia il primo giorno del mese).
WW	Il numero della settimana dell'anno (1-53, la prima settimana inizia il primo giorno dell'anno).
IW	Numero di settimana dell'anno ISO (il primo giovedì del nuovo anno è nella settimana 1.)

Parte di data o parte di ora	Significato
GIORNO, Giorno, giorno	Nome del giorno (maiuscolo, maiuscolo e minuscolo, minuscolo, riempito a vuoto con 9 caratteri)
GG, Gg, gg	Nome del giorno abbreviato (maiuscolo, maiuscolo e minuscolo, minuscolo, riempito a vuoto con 3 caratteri)
DDD	Giorno dell'anno (001-366)
IDDD	Giorno dell'anno numerazione della settimana ISO 8601 (001-371; giorno 1 dell'anno è lunedì della prima settimana ISO)
DD	Giorno del mese espresso come numero (01-31)
D	Giorno della settimana (1-7; domenica è 1) **Note** La parte di data D si comporta in modo diverso rispetto alla parte di data (DOW) utilizzata per le funzioni datetime DATE_PART e EXTRACT. DOW si basa su numeri interi 0-6, dove domenica è 0. Per ulteriori informazioni, consultare Parti di data per funzioni di data e timestamp.
ID	Giorno della settimana ISO 8601, da lunedì (1) a domenica (7)
J	Giorno giuliano (giorni trascorsi dal 1 gennaio 4712 a.C.)

Parte di data o parte di ora	Significato
HH24	Ora (formato 24 ore, 00-23)
HH o HH12	Ora (formato 12 ore, 01-12)
MI	Minuti (00-59)
SS	Secondi (00-59)
MS	Millisecondi (.000)
US	Microsecondi (.000000)
AM o PM, A.M. o P.M., a.m. o p.m., am o pm	Indicatore meridiano con distinzione tra maiuscole e minuscole (per orologio a 12 ore)
TZ, tz	Abbreviazione per il fuso orario con distinzione tra lettere maiuscole e minuscole; valido solo per TIMESTAMPTZ
OF	Compensazione da UTC; valido solo per TIMESTAMPTZ

> **Note**
>
> È necessario racchiudere i separatori di datetime (come '-', '/' o ':') in delle virgolette singole, ma si devono racchiudere le "dateparts" e "timeparts" elencate nella tabella precedente in delle virgolette doppie.

Esempi

Per esempi di formattazione di date come stringhe, consulta TO_CHAR.

Stringhe di formato numerico

Di seguito è riportato un riferimento per le stringhe in formato numerico.

Le seguenti stringhe di formato si applicano a funzioni quali TO_NUMBER e TO_CHAR.

- Per esempi di formattazione di stringhe come numeri, consulta TO_NUMBER.
- Per esempi di formattazione di numeri come stringhe, consulta TO_CHAR.

Formato	Descrizione
9	Valore numerico con il numero di cifre specificato.
0	Valore numerico con zeri iniziali.
. (periodo), D	Punto decimale.
, (virgola)	Separatore di migliaia.
CC	Codice del secolo. Ad esempio, il 21° secolo è iniziato il 01-01-2001 (supportato solo per TO_CHAR).
FM	Modalità di riempimento. Eliminare spazi vuoti e zeri.
PR	I valore negativo tra parentesi angolari.
S	Segno ancorato a un numero.
L	Simbolo di valuta nella posizione specificata.
G	Separatore di gruppo.
MI	Segno meno nella posizione specificata per numeri inferiori a 0.
PL	Segno più nella posizione specificata per numeri superiori a 0.
SG	Segno più o meno nella posizione specificata.
RN	Numero romano compreso tra 1 e 3999 (supportato solo per TO_CHAR).

Formato	Descrizione
TH o th	Suffisso del numero ordinale. Non converte numeri frazionari o valori inferiori a zero.

Caratteri di formattazione in stile Teradata per i dati numerici

Di seguito, è possibile trovare come le funzioni TEXT_TO_INT_ALT e TEXT_TO_NUMERIC_ALT interpretano i caratteri nella stringa expression di input. È inoltre possibile trovare un elenco di caratteri che è possibile specificare nella frase format. Inoltre, è possibile trovare una descrizione delle differenze tra la formattazione in stile Teradata e Amazon Redshift per l'opzione format.

Formato	Descrizione
G	Non supportato come separatore di gruppo nella stringa expression di input. Non è possibile specificare questo carattere nella frase format.
D	Simbolo radix. È possibile specificare questo carattere nella frase format. Questo carattere equivale a un . (punto). Il simbolo Radix non può essere visualizzato in una frase format che contiene uno qualsiasi dei seguenti caratteri: • . (punto) • S ('s' maiuscola) • V ('v' maiuscola)
/ , : %	Caratteri di inserimento / (barra), virgola (,), : (due punti) e % (segno di percentuale). Non è possibile includere questi caratteri nella frase format.

Formato	Descrizione
	Amazon Redshift ignora questi caratteri nella stringa expression di input.
.	Punto come carattere radix, ovvero un punto decimale. Questo carattere non può essere visualizzato in una frase format che contiene uno qualsiasi dei seguenti caratteri: • D ('d' maiuscola) • S ('s' maiuscola) • V ('v' maiuscola)
B	Non è possibile includere il carattere spazio (B) nella frase format. Nella stringa expression di input, gli spazi iniziali e finali vengono ignorati e gli spazi tra le cifre non sono consentiti.
+ -	Non è possibile includere il segno più (+) o il segno meno (-) nella frase format. Tuttavia, il segno più (+) e il segno meno (-) vengono analizzati implicitamente come parte del valore numerico se appaiono nella stringa expression di input.
V	Indicatore di posizione del punto decimale. Questo carattere non può essere visualizzato in una frase format che contiene uno qualsiasi dei seguenti caratteri: • D ('d' maiuscola) • . (punto)

Formato	Descrizione
Z	Cifra decimale con zero eliminati. Amazon Redshift taglia gli zeri iniziali. Il carattere Z non può seguire un carattere 9. Il carattere Z deve trovarsi a sinistra del carattere radix se la parte frazione contiene il carattere 9.
9	Cifra decimale.
CHAR(n)	Per questo formato, è possibile specificare: • CHAR è composto dai caratteri Z o 9. Amazon Redshift non supporta un + (più) o - (meno) nel valore CHAR. • n è una costante intera, I o F. Per I, questo è il numero di caratteri necessari per visualizz are la porzione intera di dati numerici o interi. Per F, questo è il numero di caratteri necessari per visualizzare la parte frazionaria dei dati numerici.
-	Carattere trattino (-). Non è possibile includere questo carattere nella frase format. Amazon Redshift ignora questo carattere nella stringa expression di input.

Formato	Descrizione
S	Decimale con zone firmate. Il carattere S deve seguire l'ultima cifra decimale nella frase format. L'ultimo carattere della stringa expression di input e la corrispondente conversione numerica sono riportati in Caratteri di formattazione dei dati per la formattazione dei dati numerici in stile Signed Zone Decimal, Teradata . Il carattere S non può essere visualizzato in una frase format che contiene uno qualsiasi dei seguenti caratteri: • + (segno più) • . (punto) • D ('d' maiuscola) • Z ('z' maiuscola) • F ('f' maiuscola) • E ('e' maiuscola)
E	Notazione esponenziale. La stringa expressio n di input può includere il carattere esponente . Non è possibile specificare E come carattere esponente nella frase format.
FN9	Non supportato in Amazon Redshift.
FNE	Non supportato in Amazon Redshift.

Formato	Descrizione
$, USD, Dollari USA	Il segno del dollaro ($), il simbolo di valuta ISO (USD) e il nome della valuta Dollari USA. Il simbolo di valuta ISO USD e il nome di valuta Dollari USA fanno distinzione tra maiuscole e minuscole. Amazon Redshift supporta solo la valuta USD. La stringa expression di input può includere spazi tra il simbolo di valuta USD e il valore numerico, ad esempio '$ 123E2' o '123E2 $'.
L	Simbolo di valuta. Questo simbolo di valuta può apparire una sola volta nella frase format. Non è possibile specificare più simboli di valuta.
C	Simbolo di valuta ISO. Questo simbolo di valuta può apparire una sola volta nella frase format. Non è possibile specificare più simboli di valuta.
N	Nome completo della valuta. Questo simbolo di valuta può apparire una sola volta nella frase format. Non è possibile specificare più simboli di valuta.
O	Simbolo di valuta doppio. Non è possibile specificare questo carattere nella frase format.
U	Simbolo di valuta doppio. Non è possibile specificare questo carattere nella frase format.
A	Nome completo della valuta doppio. Non è possibile specificare questo carattere nella frase format.

Caratteri di formattazione dei dati per la formattazione dei dati numerici in stile Signed Zone Decimal, Teradata

È possibile utilizzare i seguenti caratteri nella frase format delle funzioni TEXT_TO_INT_ALT e TEXT_TO_NUMERIC_ALT per un valore decimale con zone firmate.

L'ultimo carattere della stringa di input	Conversione numerica
{ o 0	n … 0
A o 1	n … 1
B o 2	n … 2
C o 3	n … 3
D o 4	n … 4
E o 5	n … 5
F o 6	n … 6
G o 7	n … 7
H o 8	n … 8
I o 9	n … 9
}	-n … 0
J	-n … 1
K	-n … 2
L	-n … 3
M	-n … 4
N	-n … 5
O	-n … 6

L'ultimo carattere della stringa di input	Conversione numerica
P	-n … 7
Q	-n … 8
R	-n … 9

Funzioni di data e ora

In questa sezione, sono riportate le informazioni sulle funzioni scalari di data e ora supportate da Amazon Redshift.

Argomenti

- Funzione LAST_DAY
- Funzione MONTHS_BETWEEN
- Funzione NEXT_DAY
- Funzione SYSDATE
- Funzione TIMEOFDAY
- Funzione TIMESTAMP_CMP
- Funzione TIMESTAMP_CMP_DATE
- Funzione TIMESTAMP_CMP_TIMESTAMPTZ
- Funzione TIMESTAMPTZ_CMP
- Funzione TIMESTAMPTZ_CMP_DATE
- Funzione TIMESTAMPTZ_CMP_TIMESTAMP
- Funzione TIMEZONE
- Funzione TO_TIMESTAMP
- Funzione TRUNC
- Parti di data per funzioni di data e timestamp

Riepilogo delle funzioni di data e ora

Funzione	Sintassi	Valori restituiti
Operatore + (concatenamento) Concatena una data a un'ora su entrambi i lati del simbolo + e restituisce un TIMESTAMPT o TIMESTAMPTZ.	data+ora	`TIMESTAMP` o `TIMESTAMPTZ`
ADD_MONTHS Aggiunge il numero di mesi specificato a una data o a un timestamp.	ADD_MONTHS ({date|timestamp}, integer)	`TIMESTAMP`
AT TIME ZONE	AT TIME ZONE 'timezone'	`TIMESTAMP` o

Funzione	Sintassi	Valori restituiti
Specifica quale fuso orario utilizzare con un'espressione TIMESTAMP o TIMESTAMP TZ.		`TIMESTAMPZ`
CONVERT_TIMEZONE Converte un timestamp da un fuso orario a un altro.	CONVERT_TIMEZONE (['timezone',] 'timezone', timestamp)	`TIMESTAMP`
CURRENT_DATE Restituisce una data nel fuso orario della sessione corrente (UTC per impostazione predefinita) per l'inizio della transazione corrente.	CURRENT_DATE	`DATE`
DATE_CMP Confronta due date e restituisce `0` se le date sono identiche, `1` se date1 è maggiore e `-1` se date2 è maggiore.	DATE_CMP (date1, date2)	`INTEGER`
DATE_CMP_TIMESTAMP Confronta una data a un'ora e restituisce `0` se i valori sono identici, `1` se date è maggiore e `-1` se timestamp è maggiore.	DATE_CMP_TIMESTAMP (date, timestamp)	`INTEGER`
DATE_CMP_TIMESTAMPTZ Confronta una data e un timestamp al fuso orario e restituisce `0` se i valori sono identici, `1` se date è maggiore e `-1` se timestamptz è maggiore.	DATE_CMP_TIMESTAMPTZ (date, timestamptz)	`INTEGER`

Funzione	Sintassi	Valori restituiti
DATE_PART_YEAR Estrae l'anno da una data.	DATE_PART_YEAR (date)	`INTEGER`
DATEADD Incrementa una data o un'ora dell'intervallo specificato.	DATEADD (datepart, interval, {date|time|timetz|timestamp})	`TIMESTAMP` o `TIME` o `TIMETZ`
DATEDIFF Restituisce la differenza tra due date o ore per una determinata parte di data, come un giorno o un mese.	DATEDIFF (datepart, {date|time|timetz|timestamp}, {date|time|timetz|timestamp})	`BIGINT`
DATE_PART Estrae un valore della parte di data da una data o un'ora.	DATE_PART (datepart, {date|timestamp})	`DOUBLE`
DATE_TRUNC Tronca un timestamp in base a una parte di data.	DATE_TRUNC ('datepart', timestamp)	`TIMESTAMP`
EXTRACT Estrae una parte di data o di ora da un timestamp, timestamptz, time o timetz.	EXTRACT (datepart FROM source)	`INTEGER or DOUBLE`
GETDATE Restituisce la data e l'ora correnti nel fuso orario della sessione corrente (UTC per impostazione predefinita). Le parentesi sono obbligatorie.	GETDATE()	`TIMESTAMP`

Funzione	Sintassi	Valori restituiti
INTERVAL_CMP Confronta due intervalli e restituisce 0 se gli intervalli sono identici, 1 se interval1 è maggiore e -1 se interval2 è maggiore.	INTERVAL_CMP (interval1, interval2)	INTEGER
LAST_DAY Restituisce la data dell'ultimo giorno del mese che contiene date.	LAST_DAY(date)	DATE
MONTHS_BETWEEN Restituisce il numero di mesi tra due date.	MONTHS_BETWEEN (date, date)	FLOAT8
NEXT_DAY Restituisce la data della prima istanza di day posteriore a date.	NEXT_DAY (date, day)	DATE
SYSDATE Restituisce la data e l'ora nel formato UTC per l'inizio della transazione corrente.	SYSDATE	TIMESTAMP
TIMEOFDAY Restituisce il giorno della settimana, la data e l'ora attuali nel fuso orario della sessione corrente (UTC per impostazione predefinita) come un valore di stringa.	TIMEOFDAY()	VARCHAR
TIMESTAMP_CMP Confronta due timestamp e restituisce 0 se i timestamp sono identici, 1 se timestamp1 è posteriore e -1 se timestamp2 è posteriore.	TIMESTAMP_CMP (timestamp1, timestamp2)	INTEGER

Funzione	Sintassi	Valori restituiti
TIMESTAMP_CMP_DATE Confronta un timestamp a una data e restituisce `0` se i valori sono identici, `1` se timestamp è maggiore e `-1` se date è maggiore.	TIMESTAMP_CMP_DATE (timestamp, date)	`INTEGER`
TIMESTAMP_CMP_TIMESTAMPTZ Confronta un timestamp a un timestamp con fuso orario e restituisce `0` se i valori sono identici, `1` se timestamp è maggiore e `-1` se timestamptz è maggiore.	TIMESTAMP_CMP_TIMESTAMPTZ (timestamp, timestamptz)	`INTEGER`
TIMESTAMPTZ_CMP Confronta due timestamp con i valori di fuso orario e restituisce `0` se i valori sono identici, `1` se timestamptz1 è maggiore e `-1` se timestamptz2 è maggiore.	TIMESTAMPTZ_CMP (timestamptz1, timestamptz2)	`INTEGER`
TIMESTAMPTZ_CMP_DATE Confronta il valore di un timestamp con fuso orario a una data e restituisce `0` se i valori sono identici, `1` se timestamptz è maggiore e `-1` se date è maggiore.	TIMESTAMPTZ_CMP_DATE (timestamptz, date)	`INTEGER`
TIMESTAMPTZ_CMP_TIMESTAMP Confronta un timestamp con fuso orario a un timestamp e restituisce `0` se i valori sono identici, `1` se timestamptz è maggiore e `-1` se timestamp è maggiore.	TIMESTAMPTZ_CMP_TIMESTAMP (timestamptz, timestamp)	`INTEGER`

Funzione	Sintassi	Valori restituiti
TIMEZONE Restituisce un timestamp per il fuso orario e il valore di timestamp specificati.	TIMEZONE ('timezone' { timestamp | timestamptz)	`TIMESTAMP` o `TIMESTAMPTZ`
TO_TIMESTAMP Restituisce un timestamp con fuso orario per il formato di timestamp e di fuso orario specificati.	TO_TIMESTAMP ('timestamp', 'format')	`TIMESTAMPTZ`
TRUNC Tronca un timestamp e restituisce una data.	TRUNC(timestamp)	`DATE`

> **Note**
>
> I secondi intercalari non vengono presi in considerazione nei calcoli del tempo trascorso.

Funzioni di data e ora nelle transazioni

Quando esegui le funzioni seguenti in un blocco di transazione (BEGIN … END), la funzione restituisce la data o l'ora di inizio della transazione corrente e non dell'istruzione corrente.

- SYSDATE
- TIMESTAMP
- CURRENT_DATE

Le funzioni seguenti restituiscono sempre la data e l'ora di inizio dell'istruzione corrente, anche quando sono in un blocco di transazione.

- GETDATE
- TIMEOFDAY

Funzioni solo sul nodo principale obsolete

Le funzioni di data seguenti sono obsolete in quanto vengono eseguite solo sul nodo principale. Per ulteriori informazioni, consulta Nodo principale: solo funzioni.

- AGE. Usare invece Funzione DATEDIFF.
- CURRENT_TIME. Utilizza invece Funzione GETDATE o SYSDATE.
- CURRENT_TIMESTAMP. Utilizza invece Funzione GETDATE o SYSDATE.
- LOCALTIME. Utilizza invece Funzione GETDATE o SYSDATE.
- LOCALTIMESTAMP. Utilizza invece Funzione GETDATE o SYSDATE.
- ISFINITE
- NOW. Utilizza invece Funzione GETDATE o SYSDATE.

Operatore + (concatenamento)

Concatena una DATA a un TIME o TIMETZ su entrambi i lati del simbolo + e restituisce un TIMESTAMPT o TIMESTAMPTZ.

Sintassi

```
date + {time | timetz}
```

L'ordine degli argomenti può essere invertito. Ad esempio, tempo + data.

Argomenti

data

Una colonna di tipo di dati `DATE` o un'espressione che restituisce un tipo `DATE`.

time

Una colonna di tipo di dati `TIME` o un'espressione che restituisce un tipo `TIME`.

timetz

Una colonna di tipo di dati `TIMETZ` o un'espressione che restituisce un tipo `TIMETZ`.

Tipo restituito

TIMESTAMP se l'input è date + time.

TIMESTAMPTZ se l'input è date + timetz.

Esempi

Configurazione di esempio

Per impostare le tabelle TIME_TEST e TIMETZ_TEST utilizzate negli esempi, utilizzare il comando seguente.

```
create table time_test(time_val time);

insert into time_test values
('20:00:00'),
('00:00:00.5550'),
('00:58:00');

create table timetz_test(timetz_val timetz);

insert into timetz_test values
('04:00:00+00'),
('00:00:00.5550+00'),
('05:58:00+00');
```

Esempi con una colonna TIME

La tabella di esempio seguente TIME_TEST contiene una colonna TIME_VAL (tipo TIME) con tre valori inseriti.

```
select time_val from time_test;

time_val
---------------------
20:00:00
00:00:00.5550
00:58:00
```

Nell'esempio seguente vengono concatenati una data letterale e una colonna TIME_VAL.

```
select date '2000-01-02' + time_val as ts from time_test;

ts
---------------------
```

```
2000-01-02 20:00:00
2000-01-02 00:00:00.5550
2000-01-02 00:58:00
```

Nell'esempio seguente vengono concatenati un valore letterale di data e un valore letterale di ora.

```
select date '2000-01-01' + time '20:00:00' as ts;

         ts
---------------------
 2000-01-01 20:00:00
```

Nell'esempio seguente vengono concatenati un valore letterale di data e un valore letterale di ora.

```
select time '20:00:00' + date '2000-01-01' as ts;

         ts
---------------------
 2000-01-01 20:00:00
```

Esempi con una colonna TIMETZ

La tabella di esempio seguente TIMETZ_TEST contiene una colonna TIMETZ_VAL (tipo TIMETZ) con tre valori inseriti.

```
select timetz_val from timetz_test;

timetz_val
------------------
04:00:00+00
00:00:00.5550+00
05:58:00+00
```

Nell'esempio seguente vengono concatenati un valore letterale di data e una colonna TIMETZ_VAL.

```
select date '2000-01-01' + timetz_val as ts from timetz_test;
ts
---------------------
2000-01-01 04:00:00+00
2000-01-01 00:00:00.5550+00
2000-01-01 05:58:00+00
```

Nell'esempio seguente vengono concatenati un valore letterale di data e una colonna TIMETZ_VAL.

```
select timetz_val + date '2000-01-01' as ts from timetz_test;
ts
--------------------
2000-01-01 04:00:00+00
2000-01-01 00:00:00.5550+00
2000-01-01 05:58:00+00
```

Nell'esempio seguente vengono concatenati un valore letterale di DATE e un valore letterale TIMETZ. L'esempio restituisce un TIMESTAMPTZ che si trova nel fuso orario UTC per impostazione predefinita. L'UTC è 8 ore avanti rispetto al PST, quindi il risultato è in anticipo di 8 ore rispetto all'ora di immissione.

```
select date '2000-01-01' + timetz '20:00:00 PST' as ts;

           ts
------------------------
 2000-01-02 04:00:00+00
```

Funzione ADD_MONTHS

ADD_MONTHS aggiunge il numero di mesi specificato a un valore o espressione di data o timestamp. La funzione DATEADD fornisce una funzionalità simile.

Sintassi

```
ADD_MONTHS( {date | timestamp}, integer)
```

Argomenti

date | timestamp

Una colonna di tipo di dati `DATE` o `TIMESTAMP` o un'espressione che implicitamente valuta un tipo `DATE` o `TIMESTAMP`. Se la data è l'ultimo giorno del mese o se il mese risultante è più corto, la funzione restituisce l'ultimo giorno del mese nel risultato. Per le altre date, il risultato contiene lo stesso numero di giorni dell'espressione di data.

integer

Un valore di tipo `INTEGER`. Utilizza un numero negativo per sottrarre mesi dalle date.

Tipo restituito

TIMESTAMP

Esempi

La query seguente utilizza la funzione ADD_MONTHS in una funzione TRUNC. La funzione TRUNC rimuove l'ora del giorno dal risultato di ADD_MONTHS. La funzione ADD_MONTHS aggiunge 12 mesi a ogni valore della colonna CALDATE. I valori nella colonna CALDATE sono date.

```
select distinct trunc(add_months(caldate, 12)) as calplus12,
trunc(caldate) as cal
from date
order by 1 asc;

 calplus12  |    cal
------------+------------
 2009-01-01 | 2008-01-01
 2009-01-02 | 2008-01-02
 2009-01-03 | 2008-01-03
...
(365 rows)
```

L'esempio seguente utilizza la funzione ADD_MONTHS per aggiungere 1 mese a un timestamp.

```
select add_months('2008-01-01 05:07:30', 1);

add_months
---------------------
2008-02-01 05:07:30
```

Gli esempi seguenti illustrano il comportamento quando la funzione ADD_MONTHS opera su date con mesi che hanno un numero di giorni differente. Questo esempio mostra come la funzione gestisce l'aggiunta di 1 mese al 31 marzo e l'aggiunta di 1 mese al 30 aprile. Aprile ha 30 giorni, quindi aggiungendo 1 mese al 31 marzo si ottiene il 30 aprile. Maggio ha 31 giorni, quindi aggiungendo 1 mese al 30 aprile si ottiene il 31 maggio.

```
select add_months('2008-03-31',1);

add_months
---------------------
```

```
2008-04-30 00:00:00

select add_months('2008-04-30',1);

add_months
---------------------
2008-05-31 00:00:00
```

Funzione AT TIME ZONE

AT TIME ZONE specifica quale fuso orario utilizzare con un'espressione TIMESTAMP o TIMESTAMPTZ.

Sintassi

```
AT TIME ZONE 'timezone'
```

Argomenti

timezone

TIMEZONE per il valore restituito. Il fuso orario può essere specificato come nome di fuso orario (ad esempio, **'Africa/Kampala'** o **'Singapore'**) oppure come abbreviazione di fuso orario (ad esempio, **'UTC'** o **'PDT'**).

Per visualizzare un elenco dei nomi di fuso orario supportati, utilizzare il comando seguente.

```
select pg_timezone_names();
```

Per visualizzare un elenco delle abbreviazioni di fuso orario supportate, utilizzare il comando seguente.

```
select pg_timezone_abbrevs();
```

Per maggiori informazioni ed esempi, consulta Note sull'utilizzo dei fusi orari.

Tipo restituito

TIMESTAMPTZ quando utilizzato con un'espressione TIMESTAMP. TIMESTAMP quando utilizzato con un'espressione TIMESTAMPTZ.

Esempi

L'esempio seguente converte un valore di timestamp senza fuso orario e lo interpreta come orario MST (UTC+7 in POSIX). L'esempio restituisce un valore di tipo di dati TIMESTAMPTZ per il fuso orario UTC. Se configuri il fuso orario predefinito su un valore diverso da UTC, potresti vedere un risultato diverso.

```
SELECT TIMESTAMP '2001-02-16 20:38:40' AT TIME ZONE 'MST';

timezone
------------------------
2001-02-17 03:38:40+00
```

Nell'esempio seguente, un timestamp di input con un valore di fuso orario EST (UTC+5 in POSIX) viene convertito in MST (UTC+7 in POSIX). L'esempio restituisce un valore di tipo di dati TIMESTAMP.

```
SELECT TIMESTAMPTZ '2001-02-16 20:38:40-05' AT TIME ZONE 'MST';

timezone
------------------------
2001-02-16 18:38:40
```

Funzione CONVERT_TIMEZONE

CONVERT_TIMEZONE converte un timestamp da un fuso orario a un altro. La funzione si regola automaticamente in base all'ora legale.

Sintassi

```
CONVERT_TIMEZONE( ['source_timezone',] 'target_timezone', 'timestamp')
```

Argomenti

source_timezone

(Facoltativo) Il fuso orario del timestamp corrente. Il valore predefinito è UTC. Per ulteriori informazioni, consultare Note sull'utilizzo dei fusi orari.

target_timezone

Il fuso orario del nuovo timestamp. Per ulteriori informazioni, consultare Note sull'utilizzo dei fusi orari.

timestamp

Una colonna timestamp o un'espressione che viene implicitamente convertita in un timestamp.

Tipo restituito

TIMESTAMP

Note sull'utilizzo dei fusi orari

source_timezone o target_timezone possono essere specificati come nome del fuso orario (come «Africa/Kampala» o «Singapore») o come abbreviazione del fuso orario (come «UTC» o «PDT»). Non è necessario convertire i nomi dei fusi orari in nomi o le abbreviazioni in abbreviazioni. Ad esempio, puoi scegliere un timestamp dal nome del fuso orario di origine «Singapore» e convertirlo in un timestamp con l'abbreviazione del fuso orario «PDT».

> **Note**
>
> I risultati dell'utilizzo del nome di un fuso orario o di un'abbreviazione del fuso orario possono essere diversi a causa dell'ora stagionale locale, ad esempio l'ora legale.

Utilizzo di un nome di fuso orario

Per visualizzare un elenco aggiornato e completo dei nomi dei fusi orari, esegui il comando seguente.

```
select pg_timezone_names();
```

Ogni riga contiene una stringa separata da virgole con il nome del fuso orario, la relativa abbreviazione, la differenza rispetto al fuso UTC e l'indicatore che mostra se il fuso orario osserva l'ora legale (`t` o `f`). Ad esempio, il frammento seguente mostra come risultati due righe. La prima riga è il fuso orario`Europe/Paris`, abbreviazione`CET`, con `01:00:00` scostamento dall'UTC e `f` per indicare che non si applica l'ora legale. La seconda riga è il fuso orario`Israel`, abbreviazione, con `02:00:00` scostamento dall'UTC`IST`, e `f` per indicarlo non si applica l'ora legale.

```
pg_timezone_names
-----------------
(Europe/Paris,CET,01:00:00,f)
(Israel,IST,02:00:00,f)
```

Esegui l'istruzione SQL per ottenere l'elenco completo e trovare il nome di un fuso orario. Vengono restituite circa 600 righe. Sebbene alcuni nomi di fusi orari restituiti siano acronimi o sigle maiuscole composte dalle lettere inziali (ad esempio, GB, PRC, ROK), la funzione CONVERT_TIMEZONE li tratta come nomi, non abbreviazioni, di fusi orari.

Se si specifica un fuso orario utilizzando un nome di fuso orario, CONVERT_TIMEZONE si adatta automaticamente all'ora legale (DST) o a qualsiasi altro protocollo stagionale locale, come l'ora legale, l'ora solare o l'ora invernale, in vigore per quel fuso orario durante la data e l'ora specificate da «timestamp». Ad esempio, "Europe/London" rappresenta UTC in inverno e aggiunge un'ora in estate.

Utilizzo di un'abbreviazione di fuso orario

Per visualizzare un elenco aggiornato e completo delle abbreviazioni dei fusi orari, esegui il comando seguente.

```
select pg_timezone_abbrevs();
```

I risultati contengono una stringa separata da virgole che include l'abbreviazione del fuso orario, la differenza rispetto al fuso UTC e l'indicatore che mostra se il fuso orario osserva l'ora legale (`t` o `f`). Ad esempio, il frammento seguente mostra come risultati due righe. La prima riga contiene l'abbreviazione dell'ora legale del Pacifico `PDT`, con una differenza di `-07:00:00` rispetto a UTC, e `t` per indicare che si osserva l'ora legale. La seconda riga contiene l'abbreviazione di Pacific Standard Time`PST`, con una differenza `-08:00:00` rispetto all'UTC e `f` per indicare che non osserva l'ora legale.

```
pg_timezone_abbrevs
--------------------
(PDT,-07:00:00,t)
(PST,-08:00:00,f)
```

Esegui l'istruzione SQL per ottenere l'elenco e completo e trovare un'abbreviazione basata sulla differenza di fuso orario e sull'indicatore relativo all'ora legale. Vengono restituite circa 200 righe.

Le abbreviazioni di fuso orario rappresentano un offset fisso rispetto a UTC. Se specificate un fuso orario utilizzando un'abbreviazione del fuso orario, CONVERT_TIMEZONE utilizza l'offset fisso rispetto all'UTC e non si adatta ad alcun protocollo stagionale locale.

Utilizzo del formato in stile POSIX

Una specifica di fuso orario in stile POSIX è in formato STDoffset o STDoffsetDST, dove STD è un'abbreviazione di fuso orario, offset è l'offset numerico in ore a ovest di UTC e DST è un'abbreviazione facoltativa di fuso orario con ora legale. L'ora legale viene considerata come un'ora avanti rispetto all'offset specificato.

I formati di fuso orario in stile POSIX utilizzano offset positivi a ovest di Greenwich, contrariamente alla convenzione ISO-8601, che utilizza offset positivi a est di Greenwich.

Di seguito sono riportati alcuni esempi di fusi orari in stile POSIX:

- PST8
- PST8PDT
- EST5
- EST5EDT

> **Note**
>
> Amazon Redshift non convalida le specifiche di fuso orario in stile POSIX, di conseguenza è possibile impostare il fuso orario su un valore non valido. Ad esempio, il comando seguente non restituisce un errore, anche se si imposta il fuso orario su un valore non valido.
>
> ```
> set timezone to ‘xxx36’;
> ```

Esempi

La maggior parte degli esempi usa il set di dati TICKIT di esempio. Per ulteriori informazioni, consulta Database di esempio.

L'esempio seguente converte il valore di timestamp dal fuso orario UTC predefinito in PST.

```
select convert_timezone('PST', '2008-08-21 07:23:54');
```

```
 convert_timezone
------------------------
2008-08-20 23:23:54
```

L'esempio seguente converte il valore di timestamp nella colonna LISTTIME dal fuso orario UTC predefinito in PST. Anche se il timestamp rientra nel periodo dell'ora legale, viene convertito nell'ora standard in quanto il fuso orario di destinazione è specificato come abbreviazione (PST).

```
select listtime, convert_timezone('PST', listtime) from listing
where listid = 16;

     listtime       |   convert_timezone
--------------------+-------------------
2008-08-24 09:36:12     2008-08-24 01:36:12
```

L'esempio seguente converte una colonna LISTTIME di timestamp dal fuso orario UTC predefinito nel fuso orario Stati Uniti/Pacifico. Il fuso orario di destinazione utilizza un nome di fuso orario e il timestamp è nel periodo dell'ora legale, di conseguenza la funzione restituisce l'ora legale.

```
select listtime, convert_timezone('US/Pacific', listtime) from listing
where listid = 16;

     listtime       |   convert_timezone
--------------------+---------------------
2008-08-24 09:36:12 | 2008-08-24 02:36:12
```

L'esempio seguente converte una stringa di timestamp da EST a PST:

```
select convert_timezone('EST', 'PST', '20080305 12:25:29');

 convert_timezone
-------------------
2008-03-05 09:25:29
```

L'esempio seguente converte un timestamp all'ora standard degli Stati Uniti orientali in quanto il fuso orario di destinazione utilizza un nome di fuso orario (America/New_York) e il timestamp si trova nel periodo dell'ora standard.

```
select convert_timezone('America/New_York', '2013-02-01 08:00:00');
```

```
 convert_timezone
---------------------
2013-02-01 03:00:00
(1 row)
```

L'esempio seguente converte il timestamp nell'ora legale dell'est degli Stati Uniti in quanto il fuso orario target utilizza un nome di fuso orario (America/New_York) e il timestamp si trova nel periodo dell'ora legale.

```
select convert_timezone('America/New_York', '2013-06-01 08:00:00');

 convert_timezone
---------------------
2013-06-01 04:00:00
(1 row)
```

L'esempio seguente illustra l'utilizzo di offset.

```
SELECT CONVERT_TIMEZONE('GMT','NEWZONE +2','2014-05-17 12:00:00') as newzone_plus_2,
CONVERT_TIMEZONE('GMT','NEWZONE-2:15','2014-05-17 12:00:00') as newzone_minus_2_15,
CONVERT_TIMEZONE('GMT','America/Los_Angeles+2','2014-05-17 12:00:00') as la_plus_2,
CONVERT_TIMEZONE('GMT','GMT+2','2014-05-17 12:00:00') as gmt_plus_2;

   newzone_plus_2    | newzone_minus_2_15  |      la_plus_2      |     gmt_plus_2
---------------------+---------------------+---------------------+---------------------
2014-05-17 10:00:00 | 2014-05-17 14:15:00 | 2014-05-17 10:00:00 | 2014-05-17 10:00:00
(1 row)
```

Funzione CURRENT_DATE

CURRENT_DATE restituisce una data nel fuso orario della sessione corrente (UTC per impostazione predefinita) nel formato predefinito: AAAA-MM-GG.

> **Note**
>
> CURRENT_DATE restituisce la data di inizio della transazione corrente e non dell'istruzione corrente. Considera lo scenario quando avvii una transazione contenente più istruzioni alle 23:59 del giorno 01/10/08 e l'istruzione contenente CURRENT_DATE viene eseguita alle 00:00 del 02/10/08. CURRENT_DATE restituisce `10/01/08`, non `10/02/08`.

Sintassi

```
CURRENT_DATE
```

Tipo restituito

DATE

Esempi

L'esempio seguente restituisce la data corrente (nel punto in cui viene eseguita la funzione). Regione AWS

```
select current_date;

   date
------------
2008-10-01
```

L'esempio seguente crea una tabella, inserisce una riga in cui l'impostazione predefinita della colonna `todays_date` è CURRENT_DATE, quindi seleziona tutte le righe della tabella.

```
CREATE TABLE insert_dates(
    label varchar(128) NOT NULL,
    todays_date DATE DEFAULT CURRENT_DATE);

INSERT INTO insert_dates(label)
VALUES('Date row inserted');

SELECT * FROM insert_dates;


 label            | todays_date
------------------+-------------
Date row inserted | 2023-05-10
```

Funzione DATE_CMP

DATE_CMP confronta due date. La funzione restituisce `0` se le date sono identiche, 1 se date1 è maggiore e `-1` se date2 è maggiore.

Sintassi

```
DATE_CMP(date1, date2)
```

Argomenti

date1

Una colonna di tipo di dati DATE o un'espressione che restituisce un tipo DATE.

date2

Una colonna di tipo di dati DATE o un'espressione che restituisce un tipo DATE.

Tipo restituito

INTEGER

Esempi

La query seguente confronta i valori DATE nella colonna CALDATE alla data del 4 gennaio 2008 e indica se il valore in CALDATE è anteriore (-1), uguale (0) o posteriore (1) al 4 gennaio 2008:

```
select caldate, '2008-01-04',
date_cmp(caldate,'2008-01-04')
from date
order by dateid
limit 10;

 caldate   |  ?column?  | date_cmp
-----------+------------+----------
2008-01-01 | 2008-01-04 |       -1
2008-01-02 | 2008-01-04 |       -1
2008-01-03 | 2008-01-04 |       -1
2008-01-04 | 2008-01-04 |        0
2008-01-05 | 2008-01-04 |        1
2008-01-06 | 2008-01-04 |        1
2008-01-07 | 2008-01-04 |        1
2008-01-08 | 2008-01-04 |        1
2008-01-09 | 2008-01-04 |        1
2008-01-10 | 2008-01-04 |        1
(10 rows)
```

Funzione DATE_CMP_TIMESTAMP

DATE_CMP_TIMESTAMP confronta un timestamp a una data e restituisce `0` se i valori sono identici, `1` se date è maggiore cronologicamente e `-1` se timestamp è maggiore.

Sintassi

```
DATE_CMP_TIMESTAMP(date, timestamp)
```

Argomenti

data

Una colonna di tipo di dati `DATE` o un'espressione che restituisce un tipo `DATE`.

timestamp

Una colonna di tipo di dati `TIMESTAMP` o un'espressione che restituisce un tipo `TIMESTAMP`.

Tipo restituito

INTEGER

Esempi

L'esempio seguente confronta la data `2008-06-18` a LISTTIME. I valori della colonna LISTTIME sono timestamp. Gli elenchi generati prima di questa data restituiscono `1`; quelli creati dopo questa data restituiscono `-1`.

```
select listid, '2008-06-18', listtime,
date_cmp_timestamp('2008-06-18', listtime)
from listing
order by 1, 2, 3, 4
limit 10;

 listid |  ?column?  |      listtime       | date_cmp_timestamp
--------+------------+---------------------+--------------------
      1 | 2008-06-18 | 2008-01-24 06:43:29 |                  1
      2 | 2008-06-18 | 2008-03-05 12:25:29 |                  1
      3 | 2008-06-18 | 2008-11-01 07:35:33 |                 -1
      4 | 2008-06-18 | 2008-05-24 01:18:37 |                  1
      5 | 2008-06-18 | 2008-05-17 02:29:11 |                  1
```

```
      6 | 2008-06-18 | 2008-08-15 02:08:13 |                -1
      7 | 2008-06-18 | 2008-11-15 09:38:15 |                -1
      8 | 2008-06-18 | 2008-11-09 05:07:30 |                -1
      9 | 2008-06-18 | 2008-09-09 08:03:36 |                -1
     10 | 2008-06-18 | 2008-06-17 09:44:54 |                 1
(10 rows)
```

Funzione DATE_CMP_TIMESTAMPTZ

DATE_CMP_TIMESTAMPTZ confronta una data e un timestamp al fuso orario e restituisce `0` se i valori sono identici, 1 se date è maggiore e `-1` se timestamptz è maggiore.

Sintassi

```
DATE_CMP_TIMESTAMPTZ(date, timestamptz)
```

Argomenti

data

Una colonna di tipo di dati `DATE` o un'espressione che restituisce un tipo `DATE`.

timestamptz

Una colonna di tipo di dati `TIMESTAMPTZ` o un'espressione che restituisce un tipo `TIMESTAMPTZ`.

Tipo restituito

INTEGER

Esempi

L'esempio seguente confronta la data `2008-06-18` a LISTTIME. Gli elenchi generati prima di questa data restituiscono 1; quelli creati dopo questa data restituiscono `-1`.

```
select listid, '2008-06-18', CAST(listtime AS timestamptz),
date_cmp_timestamptz('2008-06-18', CAST(listtime AS timestamptz))
from listing
order by 1, 2, 3, 4
limit 10;

 listid |  ?column?  |      timestamptz       | date_cmp_timestamptz
```

```
--------+------------+-------------------------+----------------------
      1 | 2008-06-18 | 2008-01-24 06:43:29+00 |                    1
      2 | 2008-06-18 | 2008-03-05 12:25:29+00 |                    1
      3 | 2008-06-18 | 2008-11-01 07:35:33+00 |                   -1
      4 | 2008-06-18 | 2008-05-24 01:18:37+00 |                    1
      5 | 2008-06-18 | 2008-05-17 02:29:11+00 |                    1
      6 | 2008-06-18 | 2008-08-15 02:08:13+00 |                   -1
      7 | 2008-06-18 | 2008-11-15 09:38:15+00 |                   -1
      8 | 2008-06-18 | 2008-11-09 05:07:30+00 |                   -1
      9 | 2008-06-18 | 2008-09-09 08:03:36+00 |                   -1
     10 | 2008-06-18 | 2008-06-17 09:44:54+00 |                    1
(10 rows)
```

Funzione DATEADD

Incrementa un valore di DATE, TIME, TIMETZ o TIMESTAMP dell'intervallo specificato.

Sintassi

```
DATEADD( datepart, interval, {date|time|timetz|timestamp} )
```

Argomenti

datepart

La parte di data (ad esempio, anno, mese o giorno) su cui la funzione opera. Per ulteriori informazioni, consultare Parti di data per funzioni di data e timestamp.

intervallo

Un intero che specifica l'intervallo (ad esempio, il numero di giorni) da aggiungere all'espressione target. Un intero negativo sottrae l'intervallo.

date|time|timetz|timestamp

Una colonna o un'espressione DATE, TIME, TIMETZ, or TIMESTAMP che viene implicitamente convertita in un DATE, TIME, TIMETZ o TIMESTAMP. L'espressione DATE, TIME, TIMETZ o TIMESTAMP deve contenere la parte di data specificata.

Tipo restituito

TIMESTAMP o TIME o TIMETZ a seconda del tipo di dati di input.

Esempi con una colonna DATE

Nel seguente esempio vengono aggiunti 30 giorni a ogni data di novembre presente nella tabella DATE.

```
select dateadd(day,30,caldate) as novplus30
from date
where month='NOV'
order by dateid;

novplus30
---------------------
2008-12-01 00:00:00
2008-12-02 00:00:00
2008-12-03 00:00:00
...
(30 rows)
```

L'esempio seguente aggiunge 18 mesi a un valore di data letterale.

```
select dateadd(month,18,'2008-02-28');

date_add
---------------------
2009-08-28 00:00:00
(1 row)
```

Il nome di colonna predefinito per una funzione DATEADD è DATE_ADD. Il timestamp predefinito per un valore di data è `00:00:00`.

Nell'esempio seguente vengono aggiunti 30 minuti a un valore di data che non specifica un timestamp.

```
select dateadd(m,30,'2008-02-28');

date_add
---------------------
2008-02-28 00:30:00
(1 row)
```

È possibile assegnare un nome completo o abbreviato alle parti di data. In questo caso, m sta per minuti, non mesi.

Esempi con una colonna TIME

La tabella di esempio seguente TIME_TEST contiene una colonna TIME_VAL (tipo TIME) con tre valori inseriti.

```
select time_val from time_test;

time_val
---------------------
20:00:00
00:00:00.5550
00:58:00
```

Nell'esempio seguente vengono aggiunti 5 minuti a ogni TIME_VAL della tabella TIME_TEST.

```
select dateadd(minute,5,time_val) as minplus5 from time_test;

minplus5
---------------
20:05:00
00:05:00.5550
01:03:00
```

Nell'esempio seguente vengono aggiunte 8 ore a un valore di tempo letterale.

```
select dateadd(hour, 8, time '13:24:55');

date_add
---------------
21:24:55
```

L'esempio seguente mostra quando un tempo va oltre 24:00:00 o è precedente a 00:00:00.

```
select dateadd(hour, 12, time '13:24:55');

date_add
---------------
01:24:55
```

Esempi con una colonna TIMETZ

I valori di output in questi esempi sono in UTC che è il fuso orario predefinito.

La tabella di esempio seguente TIMETZ_TEST contiene una colonna TIMETZ_VAL (tipo TIMETZ) con tre valori inseriti.

```
select timetz_val from timetz_test;

timetz_val
------------------
04:00:00+00
00:00:00.5550+00
05:58:00+00
```

Nell'esempio seguente vengono aggiunti 5 minuti a ogni TIMETZ_VAL nella tabella TIMETZ_TEST.

```
select dateadd(minute,5,timetz_val) as minplus5_tz from timetz_test;

minplus5_tz
---------------
04:05:00+00
00:05:00.5550+00
06:03:00+00
```

Nell'esempio seguente vengono aggiunte 2 ore a un valore timetz letterale.

```
select dateadd(hour, 2, timetz '13:24:55 PST');

date_add
---------------
23:24:55+00
```

Esempi con una colonna TIMESTAMP

I valori di output in questi esempi sono in UTC che è il fuso orario predefinito.

La tabella di esempio seguente TIMESTAMP_TEST ha una colonna TIMESTAMP_VAL (tipo TIMESTAMP) con tre valori inseriti.

```
SELECT timestamp_val FROM timestamp_test;

timestamp_val
------------------
1988-05-15 10:23:31
2021-03-18 17:20:41
```

```
2023-06-02 18:11:12
```

L'esempio seguente aggiunge solo 20 anni ai valori TIMESTAMP_VAL in TIMESTAMP_TEST prima del 2000.

```
SELECT dateadd(year,20,timestamp_val)
FROM timestamp_test
WHERE timestamp_val < to_timestamp('2000-01-01 00:00:00', 'YYYY-MM-DD HH:MI:SS');

date_add
---------------
2008-05-15 10:23:31
```

L'esempio seguente aggiunge 5 secondi a un valore di timestamp letterale scritto senza un indicatore dei secondi.

```
SELECT dateadd(second, 5, timestamp '2001-06-06');

date_add
---------------
2001-06-06 00:00:05
```

Note per l'utilizzo

Le funzioni DATEADD(month, ...) e ADD_MONTHS gestiscono differentemente le date che cadono alla fine del mese.

- ADD_MONTHS: se la data che stai aggiungendo è l'ultimo giorno del mese, il risultato è sempre l'ultimo giorno del mese risultante, indipendentemente dalla lunghezza del mese. Ad esempio, 30 aprile + 1 mese è il 31 maggio.

  ```
  select add_months('2008-04-30',1);

  add_months
  ---------------------
  2008-05-31 00:00:00
  (1 row)
  ```

- DATEADD: se vi sono meno giorni nella data che stai aggiungendo rispetto al mese del risultato, il risultato sarà il giorno corrispondente del mese risultante, non l'ultimo giorno di quel mese. Ad esempio, 30 aprile + 1 mese è il 30 maggio.

```
select dateadd(month,1,'2008-04-30');

date_add
---------------------
2008-05-30 00:00:00
(1 row)
```

La funzione DATEADD gestisce la data 29/02 dell'anno bisestile differentemente a seconda se si utilizza dateadd(month, 12,…) o dateadd(year, 1, …).

```
select dateadd(month,12,'2016-02-29');

date_add
---------------------
2017-02-28 00:00:00

select dateadd(year, 1, '2016-02-29');

date_add
---------------------
2017-03-01 00:00:00
```

Funzione DATEDIFF

DATEDIFF restituisce la differenza tra le parti di data di due espressioni di data o di ora.

Sintassi

```
DATEDIFF( datepart, {date|time|timetz|timestamp}, {date|time|timetz|timestamp} )
```

Argomenti

datepart

La parte specifica del valore di data o ora (anno, mese o giorno, ora, minuto, secondo, millisecondo o microsecondo) su cui la funzione opera. Per ulteriori informazioni, consultare Parti di data per funzioni di data e timestamp.

Più precisamente, DATEDIFF determina il numero di limiti di parte di data tra due espressioni. Ad esempio, si supponga di calcolare la differenza in anni tra due date, `12-31-2008` e

`01-01-2009`. In questo caso, la funzione restituisce 1 anno nonostante il fatto che queste date siano separate solo da un giorno. Se cerchi la differenza in ore tra due timestamp, `01-01-2009 8:30:00` e `01-01-2009 10:00:00`, il risultato è 2 ore. Se cerchi la differenza in ore tra due timestamp, `8:30:00` e `10:00:00`, il risultato è 2 ore.

date|time|timetz|timestamp

Una colonna o le espressioni DATE, TIME, TIMETZ, or TIMESTAMP che viene implicitamente convertita in un DATE, TIME, TIMETZ o TIMESTAMP. Le espressioni devono entrambe contenere la parte di data o di ora specificata. Se la seconda data o ora è posteriore alla prima, il risultato è positivo. Se la seconda data o ora è anteriore alla prima, il risultato è negativo.

Tipo restituito

BIGINT

Esempi con una colonna DATE

Nell'esempio seguente viene rilevata la differenza in numero di settimane tra due valori di data letterali.

```
select datediff(week,'2009-01-01','2009-12-31') as numweeks;

numweeks
----------
52
(1 row)
```

Nell'esempio seguente viene rilevata la differenza in ore tra due valori di data letterali. Quando non si fornisce il valore dell'ora per una data, il valore predefinito è 00:00:00.

```
select datediff(hour, '2023-01-01', '2023-01-03 05:04:03');

date_diff
----------
53
(1 row)
```

Nell'esempio seguente viene rilevata la differenza in giorni tra due valori TIMESTAMETZ letterali.

```
Select datediff(days, 'Jun 1,2008  09:59:59 EST', 'Jul 4,2008  09:59:59 EST')
```

```
date_diff
----------
33
```

Nell'esempio seguente viene rilevata la differenza, in giorni, tra due date nella stessa riga di una tabella.

```
select * from date_table;

start_date |   end_date
-----------+-----------
2009-01-01 | 2009-03-23
2023-01-04 | 2024-05-04
(2 rows)

select datediff(day, start_date, end_date) as duration from date_table;

duration
---------
      81
     486
(2 rows)
```

Nel seguente esempio viene trovata la differenza, in numero di trimestri, tra un valore letterale nel passato e la data odierna. Questo esempio presuppone che la data corrente è il 5 giugno 2008. È possibile assegnare un nome completo o abbreviato alle parti di data. Il nome di colonna predefinito per la funzione DATEDIFF è DATE_DIFF.

```
select datediff(qtr, '1998-07-01', current_date);

date_diff
-----------
40
(1 row)
```

In questo esempio viene eseguito il join delle tabelle SALES e LISTING per calcolare quanti giorni dopo la pubblicazione sono stati venduti i biglietti per i risultati da 1000 a 1005. L'attesa più lunga per la vendita di questi elenchi è di 15 giorni e quella più breve è inferiore a 1 giorno (0 giorni).

```
select priceperticket,
```

```
datediff(day, listtime, saletime) as wait
from sales, listing where sales.listid = listing.listid
and sales.listid between 1000 and 1005
order by wait desc, priceperticket desc;

priceperticket | wait
---------------+------
 96.00         |   15
 123.00        |   11
 131.00        |    9
 123.00        |    6
 129.00        |    4
 96.00         |    4
 96.00         |    0
(7 rows)
```

Questo esempio calcola il numero medio di ore di attesa dei venditori per tutte le vendite di biglietti.

```
select avg(datediff(hours, listtime, saletime)) as avgwait
from sales, listing
where sales.listid = listing.listid;

avgwait
---------
465
(1 row)
```

Esempi con una colonna TIME

La tabella di esempio seguente TIME_TEST contiene una colonna TIME_VAL (tipo TIME) con tre valori inseriti.

```
select time_val from time_test;

time_val
--------------------
20:00:00
00:00:00.5550
00:58:00
```

Nell'esempio seguente viene rilevata la differenza di numero di ore tra la colonna TIME_VAL e un valore letterale temporale.

```
select datediff(hour, time_val, time '15:24:45') from time_test;

 date_diff
-----------
        -5
        15
        15
```

Nell'esempio seguente viene rilevata la differenza in numero di minuti tra due valori letterali temporali.

```
select datediff(minute, time '20:00:00', time '21:00:00') as nummins;

nummins
----------
60
```

Esempi con una colonna TIMETZ

La tabella di esempio seguente TIMETZ_TEST contiene una colonna TIMETZ_VAL (tipo TIMETZ) con tre valori inseriti.

```
select timetz_val from timetz_test;

timetz_val
------------------
04:00:00+00
00:00:00.5550+00
05:58:00+00
```

Nell'esempio seguente vengono individuate le differenze nel numero di ore, tra un letterale TIMETZ e timetz_val.

```
select datediff(hours, timetz '20:00:00 PST', timetz_val) as numhours from timetz_test;

numhours
----------
0
-4
1
```

Nell'esempio seguente viene rilevata la differenza in numero di ore tra due valori TIMETZ letterali.

```
select datediff(hours, timetz '20:00:00 PST', timetz '00:58:00 EST') as numhours;

numhours
----------
1
```

Funzione DATE_PART

DATE_PART estrae valori di parte di data da un'espressione. DATE_PART è un sinonimo della funzione PGDATE_PART.

Sintassi

```
DATE_PART(datepart, {date|timestamp})
```

Argomenti

datepart

Un identificatore letterale o una stringa della parte specifica del valore di data (ad esempio, anno, mese o giorno) su cui la funzione opera. Per ulteriori informazioni, consulta Parti di data per funzioni di data e timestamp.

{date|timestamp}

Una colonna di data o timestamp o un'espressione che viene implicitamente convertita in una data o un timestamp. La colonna o l'espressione in data o timestamp deve contenere la parte di data specificata in datepart.

Tipo restituito

DOUBLE

Esempi

Il nome di colonna predefinito per la funzione DATE_PART è `pgdate_part`.

Per ulteriori informazioni sui dati utilizzati in alcuni degli esempi seguenti, consulta Database di esempio.

L'esempio seguente restituisce il valore di minuti da un valore di timestamp letterale.

```
SELECT DATE_PART(minute, timestamp '20230104 04:05:06.789');

pgdate_part
-----------
          5
```

Nell'esempio seguente viene rilevato il numero della settimana da un valore di timestamp letterale. Il calcolo del numero della settimana segue lo standard ISO 8601. Per ulteriori informazioni, consulta ISO 8601 in Wikipedia.

```
SELECT DATE_PART(week, timestamp '20220502 04:05:06.789');

pgdate_part
-----------
         18
```

Nell'esempio seguente viene rilevato il giorno del mese da un valore di timestamp letterale.

```
SELECT DATE_PART(day, timestamp '20220502 04:05:06.789');

pgdate_part
-----------
          2
```

Nell'esempio seguente viene rilevato il giorno della settimana da un timestamp letterale. Il numero del giorno della settimana è un numero intero compreso tra 0 e 6, iniziando da domenica.

```
SELECT DATE_PART(dayofweek, timestamp '20220502 04:05:06.789');

pgdate_part
-----------
          1
```

Nell'esempio seguente viene rilevato il secolo da un timestamp letterale. Il calcolo del secolo segue lo standard ISO 8601. Per ulteriori informazioni, consulta ISO 8601 in Wikipedia.

```
SELECT DATE_PART(century, timestamp '20220502 04:05:06.789');

pgdate_part
-----------
```

```
         21
```

L'esempio seguente restituisce il valore del millennio da un valore di timestamp letterale. Il calcolo del millennio segue lo standard ISO 8601. Per ulteriori informazioni, consulta ISO 8601 in Wikipedia.

```
SELECT DATE_PART(millennium, timestamp '20220502 04:05:06.789');

pgdate_part
-----------
         3
```

L'esempio seguente restituisce il valore di microsecondi da un valore di timestamp letterale. Il calcolo dei microsecondi segue lo standard ISO 8601. Per ulteriori informazioni, consulta ISO 8601 in Wikipedia.

```
SELECT DATE_PART(microsecond, timestamp '20220502 04:05:06.789');

pgdate_part
-----------
    789000
```

Nell'esempio seguente viene rilevato il mese da una data letterale.

```
SELECT DATE_PART(month, date '20220502');

pgdate_part
-----------
         5
```

L'esempio seguente applica la funzione DATE_PART a una colonna di una tabella.

```
SELECT date_part(w, listtime) AS weeks, listtime
FROM listing
WHERE listid=10


weeks |      listtime
------+---------------------
 25   | 2008-06-17 09:44:54
(1 row)
```

È possibile assegnare un nome completo o abbreviato alle parti di data; in questo caso, w indica settimane.

La parte di data giorno della settimana restituisce un intero da 0 a 6, partendo da domenica. Utilizza DATE_PART con dow (DAYOFWEEK) per visualizzare gli eventi di un sabato.

```
SELECT date_part(dow, starttime) AS dow, starttime
FROM event
WHERE date_part(dow, starttime)=6
ORDER BY 2,1;

 dow |      starttime
-----+---------------------
   6 | 2008-01-05 14:00:00
   6 | 2008-01-05 14:00:00
   6 | 2008-01-05 14:00:00
   6 | 2008-01-05 14:00:00
...
(1147 rows)
```

Funzione DATE_PART_YEAR

La funzione DATE_PART_YEAR estrae l'anno da una data.

Sintassi

```
DATE_PART_YEAR(date)
```

Argomento

data

Una colonna di tipo di dati `DATE` o un'espressione che restituisce un tipo `DATE`.

Tipo restituito

INTEGER

Esempi

Nell'esempio seguente viene rilevato l'anno da una data letterale.

```
SELECT DATE_PART_YEAR(date '20220502 04:05:06.789');

date_part_year
---------------
2022
```

L'esempio seguente estrae l'anno dalla colonna CALDATE: I valori nella colonna CALDATE sono date. Per ulteriori informazioni sui dati utilizzati in questo esempio, consulta Database di esempio.

```
select caldate, date_part_year(caldate)
from date
order by
dateid limit 10;

 caldate   | date_part_year
-----------+----------------
2008-01-01 |           2008
2008-01-02 |           2008
2008-01-03 |           2008
2008-01-04 |           2008
2008-01-05 |           2008
2008-01-06 |           2008
2008-01-07 |           2008
2008-01-08 |           2008
2008-01-09 |           2008
2008-01-10 |           2008
(10 rows)
```

Funzione DATE_TRUNC

La funzione DATE_TRUNC tronca un'espressione di timestamp o letterale in base alla parte di data specificata, ad esempio ora, settimana o mese.

Sintassi

```
DATE_TRUNC('datepart', timestamp)
```

Argomenti

datepart

La parte di data in corrispondenza della quale troncare il valore di timestamp. L'input timestamp viene troncato alla precisione dell'input datepart. Ad esempio, `month` viene troncato al primo giorno del mese. I formati validi sono:

- microsecond, microseconds
- millisecond, milliseconds
- second, seconds
- minute, minutes
- hour, hours
- day, days
- week, weeks
- month, months
- quarter, quarters
- year, years
- decade, decades
- century, centuries
- millennium, millennia

Per ulteriori informazioni sulle abbreviazioni di alcuni formati, consulta Parti di data per funzioni di data e timestamp

timestamp

Una colonna timestamp o un'espressione che viene implicitamente convertita in un timestamp.

Tipo restituito

TIMESTAMP

Esempi

Tronca il timestamp di input al secondo.

```
SELECT DATE_TRUNC('second', TIMESTAMP '20200430 04:05:06.789');
date_trunc
```

```
2020-04-30 04:05:06
```

Tronca il timestamp di input al minuto.

```
SELECT DATE_TRUNC('minute', TIMESTAMP '20200430 04:05:06.789');
date_trunc
2020-04-30 04:05:00
```

Tronca il timestamp di input all'ora.

```
SELECT DATE_TRUNC('hour', TIMESTAMP '20200430 04:05:06.789');
date_trunc
2020-04-30 04:00:00
```

Tronca il timestamp di input al giorno.

```
SELECT DATE_TRUNC('day', TIMESTAMP '20200430 04:05:06.789');
date_trunc
2020-04-30 00:00:00
```

Tronca il timestamp di input al primo giorno di un mese.

```
SELECT DATE_TRUNC('month', TIMESTAMP '20200430 04:05:06.789');
date_trunc
2020-04-01 00:00:00
```

Tronca il timestamp di input al primo giorno di un trimestre.

```
SELECT DATE_TRUNC('quarter', TIMESTAMP '20200430 04:05:06.789');
date_trunc
2020-04-01 00:00:00
```

Tronca il timestamp di input al primo giorno di un anno.

```
SELECT DATE_TRUNC('year', TIMESTAMP '20200430 04:05:06.789');
date_trunc
2020-01-01 00:00:00
```

Tronca il timestamp di input al primo giorno di un secolo.

```
SELECT DATE_TRUNC('millennium', TIMESTAMP '20200430 04:05:06.789');
```

```
date_trunc
2001-01-01 00:00:00
```

Tronca il timestamp di input al lunedì di una settimana.

```
select date_trunc('week', TIMESTAMP '20220430 04:05:06.789');
date_trunc
2022-04-25 00:00:00
```

Nell'esempio seguente, la funzione DATE_TRUNC utilizza la parte di data "week" per restituire la data del lunedì di ogni settimana.

```
select date_trunc('week', saletime), sum(pricepaid) from sales where
saletime like '2008-09%' group by date_trunc('week', saletime) order by 1;

date_trunc  |    sum
------------+-------------
2008-09-01  | 2474899
2008-09-08  | 2412354
2008-09-15  | 2364707
2008-09-22  | 2359351
2008-09-29  |  705249
```

Funzione EXTRACT

La funzione EXTRACT restituisce una parte della data o dell'ora da un valore TIMESTAMP, TIMESTAMPTZ, TIME, TIMETZ, INTERVAL YEAR TO MONTH o INTERVAL DAY TO SECOND. Gli esempi includono un giorno, mese, ora, minuto, secondo, millisecondo o microsecondo da un timestamp.

Sintassi

```
EXTRACT(datepart FROM source)
```

Argomenti

datepart

Il sottocampo di una data o ora da estrarre, ad esempio un giorno, un mese, un anno, un'ora, un minuto, un secondo, un millisecondo o un microsecondo. Per un elenco dei valori possibili, consultare Parti di data per funzioni di data e timestamp.

source (origine)

Colonna o espressione che restituisce un tipo di dati TIMESTAMP, TIMESTAMPTZ, TIME, TIMETZ, INTERVAL YEAR TO MONTH o INTERVAL DAY TO SECOND.

Tipo restituito

NUMERO INTERO se il valore di origine restituisce il tipo di dati TIMESTAMP, TIME, TIMETZ, INTERVAL YEAR TO MONTH o INTERVAL DAY TO SECOND.

DOUBLE PRECISION se il valore di origine restituisce il tipo di dati TIMESTAMPTZ.

Esempi con TIMESTAMP

Nel seguente esempio viene determinato il numero di settimane per le vendite il cui prezzo pagato è stato uguale o superiore a 10.000 USD. Questo esempio utilizza i dati TICKIT. Per ulteriori informazioni, consulta Database di esempio.

```
select salesid, extract(week from saletime) as weeknum
from sales
where pricepaid > 9999
order by 2;

salesid | weeknum
--------+---------
 159073 |       6
 160318 |       8
 161723 |      26
```

L'esempio seguente restituisce il valore di minuti da un valore di timestamp letterale.

```
select extract(minute from timestamp '2009-09-09 12:08:43');

date_part
-----------
8
```

L'esempio seguente restituisce il valore in millisecondi da un valore di timestamp letterale.

```
select extract(ms from timestamp '2009-09-09 12:08:43.101');
```

```
date_part
-----------
101
```

Esempi con TIMESTAMPTZ

L'esempio seguente restituisce il valore di anno da un valore di timestamp letterale.

```
select extract(year from timestamptz '1.12.1997 07:37:16.00 PST');

date_part
-----------
1997
```

Esempi con TIME

La tabella di esempio seguente TIME_TEST ha una colonna TIME_VAL (tipo TIME) con tre valori inseriti.

```
select time_val from time_test;

time_val
---------------------
20:00:00
00:00:00.5550
00:58:00
```

Nell'esempio seguente vengono estratti i minuti da ogni timetz_val.

```
select extract(minute from time_val) as minutes from time_test;

minutes
-----------
         0
         0
         58
```

Nell'esempio seguente vengono estratte le ore da ogni time_val.

```
select extract(hour from time_val) as hours from time_test;
```

```
hours
-----------
         20
         0
         0
```

Nell'esempio seguente vengono estratti i millisecondi da un valore letterale.

```
select extract(ms from time '18:25:33.123456');

 date_part
-----------
     123
```

Esempi con TIMETZ

La tabella di esempio seguente TIMETZ_TEST ha una colonna TIMETZ_VAL (tipo TIMETZ) con tre valori inseriti.

```
select timetz_val from timetz_test;

timetz_val
------------------
04:00:00+00
00:00:00.5550+00
05:58:00+00
```

Nell'esempio seguente vengono estratte le ore da ogni timez_val.

```
select extract(hour from timetz_val) as hours from time_test;

hours
-----------
         4
         0
         5
```

Nell'esempio seguente vengono estratti i millisecondi da un valore letterale. I valori letterali non vengono convertiti in UTC prima dell'elaborazione dell'estrazione.

```
select extract(ms from timetz '18:25:33.123456 EST');
```

```
 date_part
-----------
     123
```

L'esempio seguente restituisce l'ora dell'offset del fuso orario, da UTC da un valore timetz letterale.

```
select extract(timezone_hour from timetz '1.12.1997 07:37:16.00 PDT');

date_part
-----------
-7
```

Esempi con INTERVAL YEAR TO MONTH e INTERVAL DAY TO SECOND

L'esempio seguente estrae la parte relativa al giorno 1 dall'INTERVAL DAY TO SECOND che definisce 36 ore, ovvero 1 giorno e 12 ore.

```
select EXTRACT('days' from INTERVAL '36 hours' DAY TO SECOND)

 date_part
------------------
 1
```

L'esempio seguente estrae la parte relativa al mese 3 da YEAR TO MONTH che definisce 15 mesi, ovvero 1 anno e 3 mesi.

```
select EXTRACT('month' from INTERVAL '15 months' YEAR TO MONTH)

 date_part
------------------
 3
```

L'esempio seguente estrae la parte relativa al mese 6 da 30 mesi, ovvero 2 anni e 6 mesi.

```
select EXTRACT('month' from INTERVAL '30' MONTH)

 date_part
------------------
 6
```

L'esempio seguente estrae la parte oraria 2 di 50 ore, ovvero 2 giorni 2 ore.

```
select EXTRACT('hours' from INTERVAL '50' HOUR)

 date_part
------------------
 2
```

L'esempio seguente estrae la parte relativa ai minuti 11 da 1 ora e 11 minuti 11,123 secondi.

```
select EXTRACT('minute' from INTERVAL '70 minutes 70.123 seconds' MINUTE TO SECOND)

 date_part
------------------
 11
```

L'esempio seguente estrae la parte dei secondi 1.11 da 1 giorno 1 ora 1 minuto 1,11 secondi.

```
select EXTRACT('seconds' from INTERVAL '1 day 1:1:1.11' DAY TO SECOND)

 date_part
------------------
 1.11
```

L'esempio seguente estrae il numero totale di ore in un INTERVAL. Ogni parte viene estratta e aggiunta a un totale.

```
select EXTRACT('days' from INTERVAL '50' HOUR) * 24 + EXTRACT('hours' from INTERVAL
 '50' HOUR)

 ?column?
------------------
 50
```

L'esempio seguente estrae il numero totale di secondi in un INTERVAL. Ogni parte viene estratta e aggiunta a un totale.

```
select EXTRACT('days' from INTERVAL '1 day 1:1:1.11' DAY TO SECOND) * 86400 +
       EXTRACT('hours' from INTERVAL '1 day 1:1:1.11' DAY TO SECOND) * 3600 +
       EXTRACT('minutes' from INTERVAL '1 day 1:1:1.11' DAY TO SECOND) * 60 +
```

```
       EXTRACT('seconds' from INTERVAL '1 day 1:1:1.11' DAY TO SECOND)

 ?column?
------------------
 90061.11
```

Funzione GETDATE

GETDATE restituisce la data e l'ora correnti nel fuso orario della sessione corrente (UTC per impostazione predefinita). Restituisce la data o l'ora di inizio dell'istruzione corrente, anche quando si trova all'interno di un blocco di transazione.

Sintassi

```
GETDATE()
```

Le parentesi sono obbligatorie.

Tipo restituito

TIMESTAMP

Esempi

L'esempio seguente utilizza la funzione GETDATE per restituire il timestamp completo della data odierna

```
select getdate();

timestamp
---------------------
2008-12-04 16:10:43
```

L'esempio seguente utilizza la funzione GETDATE nella funzione TRUNC per restituire la data odierna senza l'ora:

```
select trunc(getdate());

trunc
------------
2008-12-04
```

Funzione INTERVAL_CMP

INTERVAL_CMP confronta due intervalli e restituisce 1 se il primo intervallo è più grande, -1 se il secondo intervallo è più grande e 0 se gli intervalli sono uguali. Per ulteriori informazioni, consultare Esempi di valori letterali a intervalli senza sintassi dei qualificatori.

Sintassi

```
INTERVAL_CMP(interval1, interval2)
```

Argomenti

interval1

Un valore di intervallo letterale.

interval2

Un valore di intervallo letterale.

Tipo restituito

INTEGER

Esempi

L'esempio seguente confronta il valore di `3 days` con `1 year`.

```
select interval_cmp('3 days','1 year');

interval_cmp
--------------
-1
```

Questo esempio confronta il valore`7 days` con `1 week`.

```
select interval_cmp('7 days','1 week');

interval_cmp
--------------
0
```

L'esempio seguente confronta il valore di `1 year` con `3 days`.

```
select interval_cmp('1 year','3 days');

interval_cmp
--------------
1
```

Funzione LAST_DAY

LAST_DAY restituisce la data dell'ultimo giorno del mese che contiene date. Il tipo di valore restituito è sempre DATE, indipendentemente dal tipo di dati dell'argomento date.

Per ulteriori informazioni sul recupero di parti di data specifiche, consulta Funzione DATE_TRUNC.

Sintassi

```
LAST_DAY( { date | timestamp } )
```

Argomenti

date | timestamp

Una colonna di tipo di dati `DATE` o `TIMESTAMP` o un'espressione che implicitamente valuta un tipo `DATE` o `TIMESTAMP`.

Tipo restituito

DATE

Esempi

L'esempio seguente restituisce la data dell'ultimo giorno del mese corrente:

```
select last_day(sysdate);

  last_day
------------
 2014-01-31
```

L'esempio seguente restituisce il numero di biglietti venduti per ognuno degli ultimi 7 giorni del mese: I valori della colonna SALETIME sono timestamp.

```
select datediff(day, saletime, last_day(saletime)) as "Days Remaining", sum(qtysold)
from sales
where datediff(day, saletime, last_day(saletime)) < 7
group by 1
order by 1;

days remaining |  sum
---------------+-------
             0 | 10140
             1 | 11187
             2 | 11515
             3 | 11217
             4 | 11446
             5 | 11708
             6 | 10988
(7 rows)
```

Funzione MONTHS_BETWEEN

MONTHS_BETWEEN determina il numero di mesi tra due date.

Se la prima data è posteriore alla seconda, il risultato è positivo, altrimenti è negativo.

Se uno degli argomenti è null, il risultato è NULL.

Sintassi

```
MONTHS_BETWEEN( date1, date2 )
```

Argomenti

date1

Una colonna di tipo di dati DATE o un'espressione che restituisce un tipo DATE.

date2

Una colonna di tipo di dati DATE o un'espressione che restituisce un tipo DATE.

Tipo restituito

FLOAT8

La parte del numero intero del risultato è basata sulla differenza tra i valori di anno e di mese delle date. La parte frazionaria del risultato viene calcolata a partire dai valori di giorno e timestamp delle date e presuppone un mese di 31 giorni.

Se date1 e date2 contengono la stessa data in un mese (ad esempio, 15/01/14 e 15/02/14) o l'ultimo giorno del mese (ad esempio, 31/08/14 e 30/09/14), il risultato è un numero intero basato sui valori di anno e mese delle date, indipendentemente dalla corrispondenza o meno dell'eventuale parte di timestamp.

Esempi

L'esempio seguente restituisce i mesi tra il 18/01/1969 e il 18/03/1969:

```
select months_between('1969-01-18', '1969-03-18')
as months;

months
----------
-2
```

L'esempio seguente restituisce i mesi tra il 18/01/1969 e il 18/01/1969:

```
select months_between('1969-01-18', '1969-01-18')
as months;

months
----------
0
```

L'esempio seguente restituisce i mesi tra la prima e l'ultima proiezione di un evento:

```
select eventname,
min(starttime) as first_show,
max(starttime) as last_show,
months_between(max(starttime),min(starttime)) as month_diff
from event
group by eventname
order by eventname
limit 5;

eventname         first_show             last_show              month_diff
-------------------------------------------------------------------------
```

```
.38 Special       2008-01-21 19:30:00.0  2008-12-25 15:00:00.0  11.12
3 Doors Down      2008-01-03 15:00:00.0  2008-12-01 19:30:00.0  10.94
70s Soul Jam      2008-01-16 19:30:00.0  2008-12-07 14:00:00.0  10.7
A Bronx Tale      2008-01-21 19:00:00.0  2008-12-15 15:00:00.0  10.8
A Catered Affair  2008-01-08 19:30:00.0  2008-12-19 19:00:00.0  11.35
```

Funzione NEXT_DAY

La funzione NEXT_DAY restituisce la data della prima istanza del giorno specificato che è posteriore alla data fornita.

Se il valore day è lo stesso giorno della settimana della data considerata, viene restituita l'occorrenza successiva di quel giorno.

Sintassi

```
NEXT_DAY( { date | timestamp }, day )
```

Argomenti

date | timestamp

Una colonna di tipo di dati `DATE` o `TIMESTAMP` o un'espressione che implicitamente valuta un tipo `DATE` o `TIMESTAMP`.

giorno

Una stringa che include il nome di qualsiasi giorno. Le maiuscole non hanno importanza.

I valori validi sono:

Day (Giorno)	Valori
Domenica	Su, Sun, Sunday
Lunedì	M, Mo, Mon, Monday
Martedì	Tu, Tue, Tues, Tuesday
Mercoledì	W, We, Wed, Wednesday
Giovedì	Th, Thu, Thurs, Thursday

Day (Giorno)	Valori
Venerdì	F, Fr, Fri, Friday
Sabato	Sa, Sat, Saturday

Tipo restituito

DATE

Esempi

L'esempio seguente restituisce la data del primo martedì dopo il 20/08/2014.

```
select next_day('2014-08-20','Tuesday');

next_day
-----------
2014-08-26
```

L'esempio seguente restituisce la data del primo martedì dopo l'1/1/2008 alle 5:54:44.

```
select listtime, next_day(listtime, 'Tue') from listing limit 1;

listtime            | next_day
--------------------+-----------
2008-01-01 05:54:44 | 2008-01-08
```

L'esempio seguente ottiene le date marketing target per il terzo trimestre:

```
select username, (firstname ||' '|| lastname) as name,
eventname, caldate, next_day (caldate, 'Monday') as marketing_target
from sales, date, users, event
where sales.buyerid = users.userid
and sales.eventid = event.eventid
and event.dateid = date.dateid
and date.qtr = 3
order by marketing_target, eventname, name;

username  |     name          |     eventname        |    caldate   |
 marketing_target
```

```
----------+------------------+---------------------+-------------
+------------------
MBO26QSG  |   Callum Atkinson | .38 Special          |  2008-07-06  | 2008-07-07
WCR50YIU  |   Erasmus Alvarez | A Doll's House       |  2008-07-03  | 2008-07-07
CKT70OIE  |   Hadassah Adkins | Ana Gabriel          |  2008-07-06  | 2008-07-07
VVG07OUO  |   Nathan Abbott   | Armando Manzanero    |  2008-07-04  | 2008-07-07
GEW77SII  |   Scarlet Avila   | August: Osage County |  2008-07-06  | 2008-07-07
ECR71CVS  |   Caryn Adkins    | Ben Folds            |  2008-07-03  | 2008-07-07
KUW82CYU  |   Kaden Aguilar   | Bette Midler         |  2008-07-01  | 2008-07-07
WZE78DJZ  |   Kay Avila       | Bette Midler         |  2008-07-01  | 2008-07-07
HXY04NVE  |   Dante Austin    | Britney Spears       |  2008-07-02  | 2008-07-07
URY81YWF  |   Wilma Anthony   | Britney Spears       |  2008-07-02  | 2008-07-07
```

Funzione SYSDATE

SYSDATE restituisce la data e l'ora correnti nel fuso orario della sessione corrente (UTC per impostazione predefinita).

> **Note**
>
> SYSDATE restituisce la data e l'ora di inizio della transazione corrente e non dell'istruzione corrente.

Sintassi

```
SYSDATE
```

Questa funzione non richiede argomenti.

Tipo restituito

TIMESTAMP

Esempi

L'esempio seguente utilizza la funzione SYSDATE per restituire il timestamp completo della data odierna.

```
select sysdate;
```

```
timestamp
----------------------------
2008-12-04 16:10:43.976353
```

L'esempio seguente utilizza la funzione SYSDATE nella funzione TRUNC per restituire la data odierna senza l'ora:

```
select trunc(sysdate);

trunc
------------
2008-12-04
```

La query seguente restituisce informazioni sulle vendite per le date comprese in un periodo a ritroso di 120 giorni a partire dalla data di esecuzione della query:

```
select salesid, pricepaid, trunc(saletime) as saletime, trunc(sysdate) as now
from sales
where saletime between trunc(sysdate)-120 and trunc(sysdate)
order by saletime asc;

 salesid | pricepaid |  saletime  |    now
---------+-----------+------------+------------
91535    |    670.00 | 2008-08-07 | 2008-12-05
91635    |    365.00 | 2008-08-07 | 2008-12-05
91901    |   1002.00 | 2008-08-07 | 2008-12-05
...
```

Funzione TIMEOFDAY

TIMEOFDAY è un alias speciale utilizzato per restituire giorno della settimana, data e ora come valore di stringa. Restituisce la stringa dell'ora del giorno per l'istruzione corrente, anche quando si trova all'interno di un blocco di transazione.

Sintassi

```
TIMEOFDAY()
```

Tipo restituito

VARCHAR

Esempi

Nell'esempio seguente viene restituita la data e l'ora correnti mediante la funzione TIMEOFDAY.

```
select timeofday();

timeofday
------------
Thu Sep 19 22:53:50.333525 2013 UTC
```

Funzione TIMESTAMP_CMP

Confronta il valore di due timestamp e restituisce un intero. Se i timestamp sono identici, la funzione restituisce `0`. Se il primo timestamp è maggiore, la funzione restituisce `1`. Se il secondo timestamp è maggiore, la funzione restituisce `-1`.

Sintassi

```
TIMESTAMP_CMP(timestamp1, timestamp2)
```

Argomenti

timestamp1

Una colonna di tipo di dati `TIMESTAMP` o un'espressione che restituisce un tipo `TIMESTAMP`.

timestamp2

Una colonna di tipo di dati `TIMESTAMP` o un'espressione che restituisce un tipo `TIMESTAMP`.

Tipo restituito

INTEGER

Esempi

Gli esempi seguenti confrontano i timestamp e mostrano i risultati del confronto.

```
SELECT TIMESTAMP_CMP('2008-01-24 06:43:29', '2008-01-24 06:43:29'),
 TIMESTAMP_CMP('2008-01-24 06:43:29', '2008-02-18 02:36:48'), TIMESTAMP_CMP('2008-02-18
 02:36:48', '2008-01-24 06:43:29');
```

```
timestamp_cmp  | timestamp_cmp | timestamp_cmp
---------------+---------------+---------------
             0 |            -1 |             1
```

L'esempio seguente confronta LISTTIME e SALETIME per un elenco. Il valore di TIMESTAMP_CMP è -1 per tutti i risultati in quanto il timestamp della vendita è successivo al timestamp del risultato:

```
select listing.listid, listing.listtime,
sales.saletime, timestamp_cmp(listing.listtime, sales.saletime)
from listing, sales
where listing.listid=sales.listid
order by 1, 2, 3, 4
limit 10;

 listid |      listtime       |      saletime       | timestamp_cmp
--------+---------------------+---------------------+---------------
      1 | 2008-01-24 06:43:29 | 2008-02-18 02:36:48 |            -1
      4 | 2008-05-24 01:18:37 | 2008-06-06 05:00:16 |            -1
      5 | 2008-05-17 02:29:11 | 2008-06-06 08:26:17 |            -1
      5 | 2008-05-17 02:29:11 | 2008-06-09 08:38:52 |            -1
      6 | 2008-08-15 02:08:13 | 2008-08-31 09:17:02 |            -1
     10 | 2008-06-17 09:44:54 | 2008-06-26 12:56:06 |            -1
     10 | 2008-06-17 09:44:54 | 2008-07-10 02:12:36 |            -1
     10 | 2008-06-17 09:44:54 | 2008-07-16 11:59:24 |            -1
     10 | 2008-06-17 09:44:54 | 2008-07-22 02:23:17 |            -1
     12 | 2008-07-25 01:45:49 | 2008-08-04 03:06:36 |            -1
(10 rows)
```

Questo esempio mostra che TIMESTAMP_CMP restituisce 0 per timestamp identici:

```
select listid, timestamp_cmp(listtime, listtime)
from listing
order by 1 , 2
limit 10;

 listid | timestamp_cmp
--------+---------------
      1 |             0
      2 |             0
      3 |             0
      4 |             0
      5 |             0
```

```
      6 |             0
      7 |             0
      8 |             0
      9 |             0
     10 |             0
(10 rows)
```

Funzione TIMESTAMP_CMP_DATE

TIMESTAMP_CMP_DATE confronta il valore di un timestamp e di una data. Se i valori di timestamp e data sono identici, la funzione restituisce `0`. Se il primo timestamp è maggiore cronologicamente, la funzione restituisce 1. Se la data è posteriore, la funzione restituisce `-1`.

Sintassi

```
TIMESTAMP_CMP_DATE(timestamp, date)
```

Argomenti

timestamp

Una colonna di tipo di dati `TIMESTAMP` o un'espressione che restituisce un tipo `TIMESTAMP`.

data

Una colonna di tipo di dati `DATE` o un'espressione che restituisce un tipo `DATE`.

Tipo restituito

INTEGER

Esempi

L'esempio seguente confronta LISTTIME e la data `2008-06-18`. Gli elenchi generati dopo questa data restituiscono `1`; quelli creati prima di questa data restituiscono `-1`. I valori LISTTIME sono timestamp.

```
select listid, listtime,
timestamp_cmp_date(listtime, '2008-06-18')
from listing
order by 1, 2, 3
```

```
limit 10;


 listid |      listtime       | timestamp_cmp_date
--------+---------------------+--------------------
      1 | 2008-01-24 06:43:29 |                 -1
      2 | 2008-03-05 12:25:29 |                 -1
      3 | 2008-11-01 07:35:33 |                  1
      4 | 2008-05-24 01:18:37 |                 -1
      5 | 2008-05-17 02:29:11 |                 -1
      6 | 2008-08-15 02:08:13 |                  1
      7 | 2008-11-15 09:38:15 |                  1
      8 | 2008-11-09 05:07:30 |                  1
      9 | 2008-09-09 08:03:36 |                  1
     10 | 2008-06-17 09:44:54 |                 -1
(10 rows)
```

Funzione TIMESTAMP_CMP_TIMESTAMPTZ

TIMESTAMP_CMP_TIMESTAMPTZ confronta il valore di un'espressione di timestamp con quello di un'espressione di timestamp con fuso orario. Se i valori di timestamp e timestamp con fuso orario sono identici, la funzione restituisce `0`. Se il primo timestamp è maggiore cronologicamente, la funzione restituisce 1. Se il timestamp con fuso orario è maggiore, la funzione restituisce `-1`.

Sintassi

```
TIMESTAMP_CMP_TIMESTAMPTZ(timestamp, timestamptz)
```

Argomenti

timestamp

Una colonna di tipo di dati `TIMESTAMP` o un'espressione che restituisce un tipo `TIMESTAMP`.

timestamptz

Una colonna di tipo di dati `TIMESTAMPTZ` o un'espressione che restituisce un tipo `TIMESTAMPTZ`.

Tipo restituito

INTEGER

Esempi

Gli esempi seguenti confrontano i timestamp con i fusi orari e mostrano i risultati del confronto.

```
SELECT TIMESTAMP_CMP_TIMESTAMPTZ('2008-01-24 06:43:29', '2008-01-24 06:43:29+00'),
 TIMESTAMP_CMP_TIMESTAMPTZ('2008-01-24 06:43:29', '2008-02-18 02:36:48+00'),
 TIMESTAMP_CMP_TIMESTAMPTZ('2008-02-18 02:36:48', '2008-01-24 06:43:29+00');

timestamp_cmp_timestamptz  | timestamp_cmp_timestamptz | timestamp_cmp_timestamptz
---------------------------+---------------------------+--------------------------
             0             |            -1             |             1
```

Funzione TIMESTAMPTZ_CMP

TIMESTAMPTZ_CMP confronta il valore di due timestamp con fuso orario e restituisce un intero. Se i timestamp sono identici, la funzione restituisce `0`. Se il primo timestamp è maggiore cronologicamente, la funzione restituisce 1. Se il secondo timestamp è maggiore, la funzione restituisce –1.

Sintassi

```
TIMESTAMPTZ_CMP(timestamptz1, timestamptz2)
```

Argomenti

timestamptz1

Una colonna di tipo di dati `TIMESTAMPTZ` o un'espressione che restituisce un tipo `TIMESTAMPTZ`.

timestamptz2

Una colonna di tipo di dati `TIMESTAMPTZ` o un'espressione che restituisce un tipo `TIMESTAMPTZ`.

Tipo restituito

INTEGER

Esempi

Gli esempi seguenti confrontano i timestamp con i fusi orari e mostrano i risultati del confronto.

```
SELECT TIMESTAMPTZ_CMP('2008-01-24 06:43:29+00', '2008-01-24 06:43:29+00'),
 TIMESTAMPTZ_CMP('2008-01-24 06:43:29+00', '2008-02-18 02:36:48+00'),
 TIMESTAMPTZ_CMP('2008-02-18 02:36:48+00', '2008-01-24 06:43:29+00');

timestamptz_cmp  | timestamptz_cmp | timestamptz_cmp
-----------------+-----------------+----------------
        0        |       -1        |       1
```

Funzione TIMESTAMPTZ_CMP_DATE

TIMESTAMPTZ_CMP_DATE confronta il valore di un timestamp e di una data. Se i valori di timestamp e data sono identici, la funzione restituisce `0`. Se il primo timestamp è maggiore cronologicamente, la funzione restituisce 1. Se la data è posteriore, la funzione restituisce `-1`.

Sintassi

```
TIMESTAMPTZ_CMP_DATE(timestamptz, date)
```

Argomenti

timestamptz

Una colonna di tipo di dati `TIMESTAMPTZ` o un'espressione che restituisce un tipo `TIMESTAMPTZ`.

data

Una colonna di tipo di dati `DATE` o un'espressione che restituisce un tipo `DATE`.

Tipo restituito

INTEGER

Esempi

L'esempio seguente confronta LISTTIME come timestamp con fuso orario con la data `2008-06-18`. Gli elenchi generati dopo questa data restituiscono 1; quelli creati prima di questa data restituiscono `-1`.

```
select listid, CAST(listtime as timestamptz) as tstz,
timestamp_cmp_date(tstz, '2008-06-18')
from listing
```

```
order by 1, 2, 3
limit 10;


 listid |          tstz          | timestamptz_cmp_date
--------+------------------------+----------------------
      1 | 2008-01-24 06:43:29+00 |                   -1
      2 | 2008-03-05 12:25:29+00 |                   -1
      3 | 2008-11-01 07:35:33+00 |                    1
      4 | 2008-05-24 01:18:37+00 |                   -1
      5 | 2008-05-17 02:29:11+00 |                   -1
      6 | 2008-08-15 02:08:13+00 |                    1
      7 | 2008-11-15 09:38:15+00 |                    1
      8 | 2008-11-09 05:07:30+00 |                    1
      9 | 2008-09-09 08:03:36+00 |                    1
     10 | 2008-06-17 09:44:54+00 |                   -1
(10 rows)
```

Funzione TIMESTAMPTZ_CMP_TIMESTAMP

TIMESTAMPTZ_CMP_TIMESTAMP confronta il valore di un'espressione di timestamp con fuso orario con quello di un'espressione di timestamp. Se i valori di timestamp con fuso orario e timestamp sono identici, la funzione restituisce `0`. Se il primo timestamp con fuso orario è maggiore cronologicamente, la funzione restituisce 1. Se il timestamp è maggiore, la funzione restituisce `-1`.

Sintassi

```
TIMESTAMPTZ_CMP_TIMESTAMP(timestamptz, timestamp)
```

Argomenti

timestamptz

Una colonna di tipo di dati `TIMESTAMPTZ` o un'espressione che restituisce un tipo `TIMESTAMPTZ`.

timestamp

Una colonna di tipo di dati `TIMESTAMP` o un'espressione che restituisce un tipo `TIMESTAMP`.

Tipo restituito

INTEGER

Esempi

Gli esempi seguenti confrontano i timestamp con i fusi orari e mostrano i risultati del confronto.

```
SELECT TIMESTAMPTZ_CMP_TIMESTAMP('2008-01-24 06:43:29+00', '2008-01-24 06:43:29'),
 TIMESTAMPTZ_CMP_TIMESTAMP('2008-01-24 06:43:29+00', '2008-02-18 02:36:48'),
 TIMESTAMPTZ_CMP_TIMESTAMP('2008-02-18 02:36:48+00', '2008-01-24 06:43:29');

timestamptz_cmp_timestamp  | timestamptz_cmp_timestamp | timestamptz_cmp_timestamp
---------------------------+---------------------------+--------------------------
             0             |            -1             |             1
```

Funzione TIMEZONE

TIMEZONE restituisce un timestamp per il fuso orario e il valore di timestamp specificati.

Per informazioni ed esempi su come impostare il fuso orario, consultare timezone.

Per informazioni ed esempi su come convertire il fuso orario, consultare CONVERT_TIMEZONE.

Sintassi

```
TIMEZONE('timezone', { timestamp | timestamptz })
```

Argomenti

timezone

Il fuso orario del valore restituito. Il fuso orario può essere specificato come nome di fuso orario (ad esempio, **'Africa/Kampala'** o **'Singapore'**) oppure come abbreviazione di fuso orario (ad esempio, **'UTC'** o **'PDT'**). Per visualizzare un elenco dei nomi di fuso orario supportati, utilizzare il comando seguente.

```
select pg_timezone_names();
```

Per visualizzare un elenco delle abbreviazioni di fuso orario supportate, utilizzare il comando seguente.

```
select pg_timezone_abbrevs();
```

Per maggiori informazioni ed esempi, consulta Note sull'utilizzo dei fusi orari.

timestamp | timestamptz

Un'espressione che restituisce un tipo TIMESTAMP o TIMESTAMPTZ o un valore che può essere implicitamente convertito in un timestamp o in un timestamp con fuso orario.

Tipo restituito

TIMESTAMPTZ quando utilizzato con un'espressione TIMESTAMP.

TIMESTAMP quando utilizzato con un'espressione TIMESTAMPTZ.

Esempi

Quanto segue restituisce un timestamp per il fuso orario UTC utilizzando il timestamp `2008-06-17 09:44:54` dal fuso orario PST.

```
SELECT TIMEZONE('PST', '2008-06-17 09:44:54');

timezone
----------------------
2008-06-17 17:44:54+00
```

Quanto segue restituisce un timestamp per il fuso orario PST utilizzando il timestamp `2008-06-17 09:44:54+00` dal fuso orario UTC.

```
SELECT TIMEZONE('PST', timestamptz('2008-06-17 09:44:54+00'));

timezone
----------------------
2008-06-17 01:44:54
```

Funzione TO_TIMESTAMP

TO_TIMESTAMP converte una stringa TIMESTAMP in TIMESTAMPTZ. Per un elenco di funzioni aggiuntive di data e ora per Amazon Redshift, consulta Funzioni di data e ora.

Sintassi

```
to_timestamp(timestamp, format)
```

```
to_timestamp (timestamp, format, is_strict)
```

Argomenti

timestamp

Una stringa che rappresenta un valore timestamp nel formato specificato da format. Se questo argomento viene lasciato vuoto, il valore del timestamp predefinito è `0001-01-01 00:00:00`.

format

Una letterale di stringa che definisce il formato del valore timestamp. I formati che includono un fuso orario (**`TZ`**, **`tz`** o **`OF`**) non sono supportati come input. Per i formati di timestamp validi, consultare Stringhe di formato datetime.

is_strict

Un valore booleano facoltativo che specifica se viene restituito un errore se un valore timestamp di input non è compreso nell'intervallo. Quando is_strict è impostato su TRUE, viene restituito un errore se esiste un valore fuori intervallo. Quando is_strict è impostato su FALSE, che è il valore di default, allora i valori di overflow sono accettati.

Tipo restituito

TIMESTAMPTZ

Esempi

L'esempio seguente mostra l'utilizzo della funzione TO_TIMESTAMP per convertire una stringa TIMESTAMP in TIMESTAMPTZ.

```
select sysdate, to_timestamp(sysdate, 'YYYY-MM-DD HH24:MI:SS') as second;

timestamp                  | second
---------------------------   ---------------------
2021-04-05 19:27:53.281812 | 2021-04-05 19:27:53+00
```

È possibile passare a TO_TIMESTAMP parte di una data. Le parti rimanenti della data sono impostate sui valori predefiniti. L'orario è incluso nell'output:

```
SELECT TO_TIMESTAMP('2017','YYYY');
```

```
to_timestamp
--------------------------
2017-01-01 00:00:00+00
```

L'istruzione SQL seguente converte la stringa '2011-12-18 24:38:15' in un TIMESTAMPTZ. Il risultato è un TIMESTAMPTZ che cade il giorno successivo perché il numero di ore è superiore a 24 ore:

```
SELECT TO_TIMESTAMP('2011-12-18 24:38:15', 'YYYY-MM-DD HH24:MI:SS');

to_timestamp
---------------------
2011-12-19 00:38:15+00
```

L'istruzione SQL seguente converte la stringa '2011-12-18 24:38:15' in un TIMESTAMPTZ. Il risultato è un errore perché il valore dell'ora nel timestamp è superiore a 24 ore:

```
SELECT TO_TIMESTAMP('2011-12-18 24:38:15', 'YYYY-MM-DD HH24:MI:SS', TRUE);

ERROR:  date/time field time value out of range: 24:38:15.0
```

Funzione TRUNC

Tronca un `TIMESTAMP` e restituisce `DATE`.

Questa funzione può anche troncare un numero. Per ulteriori informazioni, consulta Funzione TRUNC.

Sintassi

```
TRUNC(timestamp)
```

Argomenti

timestamp

Una colonna di tipo di dati `TIMESTAMP` o un'espressione che restituisce un tipo `TIMESTAMP`.

Per restituire un valore di timestamp con come ora, eseguire il casting del risultato della funzione su `TIMESTAMP`.

Tipo restituito

DATE

Esempi

Nel seguente esempio viene restituita la parte di data dal risultato della funzione SYSDATE (che restituisce un timestamp).

```
SELECT SYSDATE;

+----------------------------+
|         timestamp          |
+----------------------------+
| 2011-07-21 10:32:38.248109 |
+----------------------------+

SELECT TRUNC(SYSDATE);

+------------+
|   trunc    |
+------------+
| 2011-07-21 |
+------------+
```

Nell'esempio seguente viene applicata la funzione TRUNC a una colonna `TIMESTAMP`. Il tipo restituito è una data.

```
SELECT TRUNC(starttime) FROM event
ORDER BY eventid LIMIT 1;

+------------+
|   trunc    |
+------------+
| 2008-01-25 |
+------------+
```

L'esempio seguente restituisce un valore di timestamp con `00:00:00` come ora trasmessa dal risultato della funzione TRUNC su `TIMESTAMP`.

```
SELECT CAST((TRUNC(SYSDATE)) AS TIMESTAMP);
```

```
+---------------------+
|        trunc        |
+---------------------+
| 2011-07-21 00:00:00 |
+---------------------+
```

Parti di data per funzioni di data e timestamp

La tabella seguente identifica i nomi e le abbreviazioni di parti di data e parti di ora accettati come argomenti per le seguenti funzioni:

- DATEADD
- DATEDIFF
- DATE_PART
- EXTRACT

Parte data o parte ora	Abbreviazioni
millennium, millennia	mil, mils
century, centuries	c, cent, cents
decade, decades	dec, decs
epoch	epoca (supportato da EXTRACT)
year, years	y, yr, yrs
quarter, quarters	qtr, qtrs
month, months	mon, mons
week, weeks	w
day of week	dayofweek, dow, dw, weekday (supportate da DATE_PART e Funzione EXTRACT) Restituisce un intero compreso tra 0 e 6, a partire da domenica.

Parte data o parte ora	Abbreviazioni
	Note La parte di data DOW si comporta in modo diverso rispetto alla parte di data day of week (D) utilizzata per stringhe in formato datetime. D si basa su numeri interi 1-7, dove domenica è 1. Per ulteriori informazioni, consultare Stringhe di formato datetime.
day of year	dayofyear, doy, dy, yearday (supportato da EXTRACT)
day, days	d
hour, hours	h, hr, hrs
minute, minutes	m, min, mins
second, seconds	s, sec, secs
millisecond, milliseconds	ms, msec, msecs, msecond, mseconds, millisec, millisecs, millisecon
microsecond, microseconds	microsec, microsecs, microsecond, usecond, useconds, us, usec, usecs
timezone, timezone_hour, timezone_minute	Supportato da EXTRACT solo per il timestamp con fuso orario (TIMESTAMPTZ).

Variazioni nei risultati con secondi, millisecondi e microsecondi

Differenze minori nei risultati delle query si hanno quando funzioni di data differenti specificano secondi, millisecondi o microsecondi come parti di data:

- La funzione EXTRACT restituisce interi solo per la parte di data specificata, ignorando parti di dati di livello superiore e inferiore. Se la parte di data specificata è secondi, millisecondi e microsecondi non sono inclusi nel risultato. Se la parte di data specificata è millisecondi, secondi e microsecondi non sono inclusi nel risultato. Se la parte di data specificata è microsecondi, secondi e millisecondi non sono inclusi nel risultato.

- La funzione DATE_PART restituisce la parte di secondi completa del timestamp, indipendentemente dalla parte di data specificata, restituendo un valore decimale o un intero in base alle necessità.

Ad esempio, confronta i risultati delle seguenti query:

```
create table seconds(micro timestamp);

insert into seconds values('2009-09-21 11:10:03.189717');

select extract(sec from micro) from seconds;

date_part
-----------
3

select date_part(sec, micro) from seconds;

pgdate_part
-------------
3.189717
```

Note su CENTURY, EPOCH, DECADE e MIL

CENTURY o CENTURIES

Amazon Redshift interpreta CENTURY con inizio nell'anno ###1 e fine nell'anno ###0:

```
select extract (century from timestamp '2000-12-16 12:21:13');
date_part
-----------
20

select extract (century from timestamp '2001-12-16 12:21:13');
date_part
-----------
21
```

EPOCA

L'implementazione di EPOCH in Amazon Redshift è relativa a 1970-01-01 00:00:00.000000 indipendentemente del fuso orario in cui si trova il cluster. È possibile che sia necessario compensare i risultati della differenza in ore a seconda del fuso orario in cui si trova il cluster.

L'esempio seguente mostra quanto segue:

1. Crea una tabella denominata EVENT_EXAMPLE in funzione della tabella EVENT. Questo comando CREATE AS utilizza la funzione DATE_PART per creare una colonna data (denominata PGDATE_PART per impostazione predefinita) e archiviare il valore epoch per ogni evento.
2. Seleziona la colonna e il tipo di dati di EVENT_EXAMPLE da PG_TABLE_DEF.
3. Seleziona EVENTNAME, STARTTIME e PGDATE_PART dalla tabella EVENT_EXAMPLE per visualizzare i differenti formati di data e ora.
4. Seleziona EVENTNAME e STARTTIME da EVENT EXAMPLE così com'è. Converte i valori epoch in PGDATE_PART utilizzando un intervallo di un secondo per un timestamp senza fuso orario e restituisce i risultati in una colonna denominata CONVERTED_TIMESTAMP.

```
create table event_example
as select eventname, starttime, date_part(epoch, starttime) from event;

select "column", type from pg_table_def where tablename='event_example';

     column    |            type
---------------+-----------------------------
 eventname     | character varying(200)
 starttime     | timestamp without time zone
 pgdate_part   | double precision
(3 rows)
```

```
select eventname, starttime, pgdate_part from event_example;

   eventname          |      starttime      | pgdate_part
----------------------+---------------------+-------------
 Mamma Mia!           | 2008-01-01 20:00:00 |  1199217600
 Spring Awakening     | 2008-01-01 15:00:00 |  1199199600
 Nas                  | 2008-01-01 14:30:00 |  1199197800
 Hannah Montana       | 2008-01-01 19:30:00 |  1199215800
 K.D. Lang            | 2008-01-01 15:00:00 |  1199199600
```

```
 Spamalot             | 2008-01-02 20:00:00 |  1199304000
 Macbeth              | 2008-01-02 15:00:00 |  1199286000
 The Cherry Orchard   | 2008-01-02 14:30:00 |  1199284200
 Macbeth              | 2008-01-02 19:30:00 |  1199302200
 Demi Lovato          | 2008-01-02 19:30:00 |  1199302200


select eventname,
starttime,
timestamp with time zone 'epoch' + pgdate_part * interval '1 second' AS
 converted_timestamp
from event_example;

       eventname      |      starttime      | converted_timestamp
----------------------+---------------------+---------------------
 Mamma Mia!           | 2008-01-01 20:00:00 | 2008-01-01 20:00:00
 Spring Awakening     | 2008-01-01 15:00:00 | 2008-01-01 15:00:00
 Nas                  | 2008-01-01 14:30:00 | 2008-01-01 14:30:00
 Hannah Montana       | 2008-01-01 19:30:00 | 2008-01-01 19:30:00
 K.D. Lang            | 2008-01-01 15:00:00 | 2008-01-01 15:00:00
 Spamalot             | 2008-01-02 20:00:00 | 2008-01-02 20:00:00
 Macbeth              | 2008-01-02 15:00:00 | 2008-01-02 15:00:00
 The Cherry Orchard   | 2008-01-02 14:30:00 | 2008-01-02 14:30:00
 Macbeth              | 2008-01-02 19:30:00 | 2008-01-02 19:30:00
 Demi Lovato          | 2008-01-02 19:30:00 | 2008-01-02 19:30:00
 ...
```

DECADE o DECADES

Amazon Redshift interpreta DECADE o DECADES DATEPART in base al calendario comune. Ad esempio, poiché il calendario comune inizia dall'anno 1, il primo decennio (decennio 1) va da 0001-01-01 a 0009-12-31 e il secondo decennio (decennio 2) va da 0010-01-01 a 0019-12-31. Ad esempio, il decennio 201 va da 2000-01-01 a 2009-12-31:

```
select extract(decade from timestamp '1999-02-16 20:38:40');
date_part
-----------
200

select extract(decade from timestamp '2000-02-16 20:38:40');
date_part
-----------
201
```

```
select extract(decade from timestamp '2010-02-16 20:38:40');
date_part
-----------
202
```

MIL o MILS

Amazon Redshift interpreta MIL con inizio il primo giorno dell'anno #001 e fine l'ultimo giorno dell'anno #000:

```
select extract (mil from timestamp '2000-12-16 12:21:13');
date_part
-----------
2

select extract (mil from timestamp '2001-12-16 12:21:13');
date_part
-----------
3
```

Funzioni hash

Argomenti

- Funzione CHECKSUM
- Funzione farmFingerprint64
- Funzione FUNC_SHA1
- Funzione FNV_HASH
- Funzione MD5
- Funzione SHA
- Funzione SHA1
- Funzione SHA2
- MURMUR3_32_HASH

Una funzione hash è una funzione matematica che converte un valore di input numerico in un altro valore.

Funzione CHECKSUM

Calcola un valore di checksum per costruire un indice di hash.

Sintassi

```
CHECKSUM(expression)
```

Argomento

espressione

L'espressione di input deve essere un tipo di dati VARCHAR, INTEGER o DECIMAL.

Tipo restituito

La funzione CHECKSUM restituisce un integer.

Esempio

L'esempio seguente calcola un valore di checksum per la colonna COMMISSION:

```
select checksum(commission)
from sales
order by salesid
limit 10;

checksum
----------
10920
1140
5250
2625
2310
5910
11820
2955
8865
975
(10 rows)
```

Funzione farmFingerprint64

Calcola il valore farmhash dell'argomento di input utilizzando la funzione `Fingerprint64`.

Sintassi

```
farmFingerprint64(expression)
```

Argomento

espressione

L'espressione di input deve essere un tipo di dati `VARCHAR` o `VARBYTE`.

Tipo restituito

La funzione `farmFingerprint64` restituisce `BIGINT`.

Esempio

Nell'esempio seguente viene restituito il valore `farmFingerprint64` di `Amazon Redshift` che è inserito come tipo di dati `VARCHAR`.

```
SELECT farmFingerprint64('Amazon Redshift');
```

```
  farmfingerprint64
---------------------
 8085098817162212970
```

Nell'esempio seguente viene restituito il valore `farmFingerprint64` di `Amazon Redshift` che è inserito come tipo di dati `VARBYTE`.

```
SELECT farmFingerprint64('Amazon Redshift'::varbyte);
```

```
  farmfingerprint64
---------------------
```

```
8085098817162212970
```

Funzione FUNC_SHA1

Sinonimo della funzione SHA1.

Per informazioni, consultare Funzione SHA1.

Funzione FNV_HASH

Calcola la funzione hash non crittografica FNV-1a a 64 bit per tutti i tipi di dati di base.

Sintassi

```
FNV_HASH(value [, seed])
```

Argomenti

value

Il valore di input da sottoporre a hash. Amazon Redshift utilizza la rappresentazione binaria del valore per sottoporre a hash il valore di input; ad esempio, i valori INTEGER sono sottoposti a hash utilizzando 4 byte e i valori BIGINT sono sottoposti a hash utilizzando 8 byte. Inoltre, l'hash di input CHAR e VARCHAR non ignora gli spazi finali.

seed

Il seed BIGINT della funzione hash è facoltativo. Se non viene fornito, Amazon Redshift utilizza il seed FNV di default. Ciò consente di combinare l'hash di più colonne senza conversioni o concatenazioni.

Tipo restituito

BIGINT

Esempio

Gli esempi seguenti restituiscono l'hash FNV di un numero, la stringa "Amazon Redshift" e la concatenazione dei due.

```
select fnv_hash(1);
```

```
        fnv_hash
----------------------
 -5968735742475085980
(1 row)
```

```
select fnv_hash('Amazon Redshift');
      fnv_hash
--------------------
 7783490368944507294
(1 row)
```

```
select fnv_hash('Amazon Redshift', fnv_hash(1));
       fnv_hash
----------------------
 -2202602717770968555
(1 row)
```

Note per l'utilizzo

- Per calcolare l'hash di una tabella con più colonne, puoi calcolare l'hash FNV della prima colonna e passarlo come seed all'hash della seconda colonna. Quindi, passa l'hash FNV della seconda colonna come seed all'hash della terza colonna.

  L'esempio seguente crea i seed per sottoporre all'hash una tabella con più colonne.

  ```
  select fnv_hash(column_3, fnv_hash(column_2, fnv_hash(column_1))) from sample_table;
  ```

- La stessa proprietà può essere utilizzata per calcolare l'hash di una concatenazione di stringhe.

  ```
  select fnv_hash('abcd');
         fnv_hash
  ----------------------
   -281581062704388899
  (1 row)
  ```

  ```
  select fnv_hash('cd', fnv_hash('ab'));
        fnv_hash
  ----------------------
   -281581062704388899
  (1 row)
  ```

- La funzione hash utilizza il tipo di input per determinare il numero di byte da sottoporre all'hash. Utilizzare il casting per applicare un tipo specifico, se necessario.

  Negli esempi seguenti vengono utilizzati diversi tipi di input per produrre risultati diversi.

```
select fnv_hash(1::smallint);
      fnv_hash
--------------------
 589727492704079044
(1 row)
```

```
select fnv_hash(1);
       fnv_hash
----------------------
 -5968735742475085980
(1 row)
```

```
select fnv_hash(1::bigint);
       fnv_hash
----------------------
 -8517097267634966620
(1 row)
```

Funzione MD5

Utilizza la funzione di hash crittografica MD5 per convertire una stringa di lunghezza variabile in una stringa di 32 caratteri che è una rappresentazione testuale del valore esadecimale di un checksum a 128 bit.

Sintassi

```
MD5(string)
```

Argomenti

stringa

Una stringa di lunghezza variabile.

Tipo restituito

La funzione MD5 restituisce una stringa di 32 caratteri che è una rappresentazione testuale del valore esadecimale di un checksum a 128 bit.

Esempi

L'esempio seguente mostra il valore a 128 bit per la stringa 'Amazon Redshift':

```
select md5('Amazon Redshift');
md5
----------------------------------
f7415e33f972c03abd4f3fed36748f7a
(1 row)
```

Funzione SHA

Sinonimo della funzione SHA1.

Per informazioni, consultare Funzione SHA1.

Funzione SHA1

La funzione SHA1 utilizza la funzione di hash crittografica SHA1 per convertire una stringa di lunghezza variabile in una stringa di 40 caratteri che è una rappresentazione testuale del valore esadecimale di un checksum a 160 bit.

Sintassi

SHA1 è un sinonimo di Funzione SHA e Funzione FUNC_SHA1.

```
SHA1(string)
```

Argomenti

stringa

Una stringa di lunghezza variabile.

Tipo restituito

La funzione SHA1 restituisce una stringa di 40 caratteri che è una rappresentazione testuale del valore esadecimale di un checksum a 160 bit.

Esempio

L'esempio seguente restituisce il valore a 160 bit per la parola 'Amazon Redshift':

```
select sha1('Amazon Redshift');
```

Funzione SHA2

La funzione SHA2 utilizza la funzione di hash crittografica SHA2 per convertire una stringa di lunghezza variabile in una stringa di caratteri. La stringa di caratteri è una rappresentazione testuale del valore esadecimale del checksum con il numero specificato di bit.

Sintassi

```
SHA2(string, bits)
```

Argomenti

stringa

Una stringa di lunghezza variabile.

integer

Numero di bit nelle funzioni hash. I valori validi sono 0 (uguale a 256), 224, 256, 384 e 512.

Tipo restituito

La funzione SHA2 restituisce una stringa di caratteri che è una rappresentazione testuale del valore esadecimale del checksum o una stringa vuota se il numero di bit non è valido.

Esempio

L'esempio seguente restituisce il valore a 256 bit per la parola 'Amazon Redshift':

```
select sha2('Amazon Redshift', 256);
```

MURMUR3_32_HASH

La funzione MURMUR3_32_HASH calcola l'hash non crittografato Murmur3A a 32 bit per tutti i tipi di dati comuni, inclusi i tipi numerici e di stringa.

Sintassi

```
MURMUR3_32_HASH(value [, seed])
```

Argomenti

value

Il valore di input da sottoporre a hash. Amazon Redshift esegue l'hash della rappresentazione binaria del valore di input. Questo comportamento è simile a Funzione FNV_HASH, ma il valore viene convertito nella rappresentazione binaria specificata dalla specifica hash Murmur3 a 32 bit di Apache Iceberg.

seed

Il seed INT della funzione hash. Questo argomento è facoltativo. Se non viene fornito, Amazon Redshift utilizza il seed predefinito 0. Ciò consente di combinare l'hash di più colonne senza conversioni o concatenazioni.

Tipo restituito

La funzione restituisce un INT.

Esempio

Gli esempi seguenti restituiscono l'hash Murmur3 di un numero, la stringa "Amazon Redshift" e la concatenazione dei due.

```
select MURMUR3_32_HASH(1);

    MURMUR3_32_HASH
----------------------
 -596735742475085980
(1 row)
```

```
select MURMUR3_32_HASH('Amazon Redshift');
```

```
    MURMUR3_32_HASH
----------------------
 7783490368944507294
(1 row)
```

```
select MURMUR3_32_HASH('Amazon Redshift', MURMUR3_32_HASH(1));

    MURMUR3_32_HASH
----------------------
 -2202602717770968555
(1 row)
```

Note per l'utilizzo

Per calcolare l'hash di una tabella con più colonne, puoi calcolare l'hash Murmur3 della prima colonna e passarlo come seed all'hash della seconda colonna. Quindi, passa l'hash Murmur3 della seconda colonna come seed all'hash della terza colonna.

L'esempio seguente crea i seed per sottoporre all'hash una tabella con più colonne.

```
select MURMUR3_32_HASH(column_3, MURMUR3_32_HASH(column_2, MURMUR3_32_HASH(column_1)))
 from sample_table;
```

La stessa proprietà può essere utilizzata per calcolare l'hash di una concatenazione di stringhe.

```
select MURMUR3_32_HASH('abcd');

   MURMUR3_32_HASH
----------------------
 -281581062704388899
(1 row)
```

```
select MURMUR3_32_HASH('cd', MURMUR3_32_HASH('ab'));

   MURMUR3_32_HASH
----------------------
 -281581062704388899
(1 row)
```

La funzione hash utilizza il tipo di input per determinare il numero di byte da sottoporre all'hash. Utilizzare il casting per applicare un tipo specifico, se necessario.

Negli esempi seguenti vengono utilizzati diversi tipi di input per produrre risultati differenti.

```
select MURMUR3_32_HASH(1::smallint);

   MURMUR3_32_HASH
--------------------
 589727492704079044
(1 row)
```

```
select MURMUR3_32_HASH(1);

   MURMUR3_32_HASH
----------------------
 -5968735742475085980
(1 row)
```

```
select MURMUR3_32_HASH(1::bigint);

   MURMUR3_32_HASH
----------------------
 -8517097267634966620
(1 row)
```

HyperLogLog funzioni

Di seguito, puoi trovare le descrizioni delle HyperLogLog funzioni per SQL supportate da Amazon Redshift.

Argomenti

- Funzione HLL
- Funzione HLL_CREATE_SKETCH
- Funzione HLL_CARDINALITY
- Funzione HLL_COMBINE
- Funzione HLL_COMBINE_SKETCHES

Funzione HLL

La funzione HLL restituisce la HyperLogLog cardinalità dei valori delle espressioni di input. La funzione HLL funziona con qualsiasi tipo di dati ad eccezione del tipo di dati HLLSKETCH. La funzione HLL ignora i valori NULL. Quando non ci sono righe in una tabella o tutte le righe sono NULL, la cardinalità risultante è 0.

Sintassi

```
HLL (aggregate_expression)
```

Argomento

aggregate_expression

Qualsiasi espressione valida (come il nome di una colonna) che fornisce i valori da aggregare. Questa funzione supporta qualsiasi tipo di dati come input tranne HLLSKETCH, GEOMETRY, GEOGRAPHY e VARBYTE.

Tipo restituito

La funzione HLL restituisce un valore BIGINT o INT8.

Esempi

L'esempio seguente restituisce la cardinalità della colonna `an_int` nella tabella `a_table`.

```
CREATE TABLE a_table(an_int INT);
INSERT INTO a_table VALUES (1), (2), (3), (4);

SELECT hll(an_int) AS cardinality FROM a_table;
cardinality
-------------
4
```

Funzione HLL_CREATE_SKETCH

La funzione HLL_CREATE_SKETCH restituisce un tipo di dati HLLSKETCH che incapsula i valori delle espressioni di input. La funzione HLL_CREATE_SKETCH funziona con qualsiasi tipo di dati

e ignora i valori NULL. Quando non ci sono righe in una tabella o tutte le righe sono NULL, lo schizzo risultante non ha coppie indice-valore come `{"version":1,"logm":15,"sparse":{"indices":[],"values":[]}}`.

Sintassi

```
HLL_CREATE_SKETCH (aggregate_expression)
```

Argomento

aggregate_expression

Qualsiasi espressione valida (come il nome di una colonna) che fornisce i valori da aggregare. I valori NULL vengono ignorati. Questa funzione supporta qualsiasi tipo di dati come input tranne HLLSKETCH, GEOMETRY, GEOGRAPHY e VARBYTE.

Tipo restituito

La funzione HLL_CREATE_SKETCH restituisce un valore HLLSKETCH.

Esempi

L'esempio seguente restituisce il tipo HLLSKETCH per la colonna `an_int` nella tabella `a_table`. Un oggetto JSON viene utilizzato per rappresentare uno schizzo sparso durante l'importazione, l'esportazione o la stampa di HyperLogLog schizzi. Una rappresentazione di stringa (in formato Base64) viene utilizzata per rappresentare uno schizzo denso. HyperLogLog

```
CREATE TABLE a_table(an_int INT);
INSERT INTO a_table VALUES (1), (2), (3), (4);

SELECT hll_create_sketch(an_int) AS sketch FROM a_table;
sketch
-----------------------------------------------------------------------------------------------
{"version":1,"logm":15,"sparse":{"indices":
[20812342,20850007,22362299,47158030],"values":[1,2,1,1]}}
(1 row)
```

Funzione HLL_CARDINALITY

La funzione HLL_CARDINALITY restituisce la cardinalità del tipo di dati HLLSKETCH di input.

Sintassi

```
HLL_CARDINALITY (hllsketch_expression)
```

Argomento

hllsketch_expression

Qualsiasi espressione valida che restituisce un tipo HLLSKETCH, come ad esempio un nome di colonna. Il valore di input è il tipo di dati HLLSKETCH.

Tipo restituito

La funzione HLL_CARDINALITY restituisce un valore BIGINT o INT8.

Esempi

L'esempio seguente restituisce la cardinalità della colonna `sketch` nella tabella `hll_table`.

```
CREATE TABLE a_table(an_int INT, b_int INT);
INSERT INTO a_table VALUES (1,1), (2,1), (3,1), (4,1), (1,2), (2,2), (3,2), (4,2),
 (5,2), (6,2);

CREATE TABLE hll_table (sketch HLLSKETCH);
INSERT INTO hll_table select hll_create_sketch(an_int) from a_table group by b_int;

SELECT hll_cardinality(sketch) AS cardinality FROM hll_table;
cardinality
-------------
6
4
(2 rows)
```

Funzione HLL_COMBINE

La funzione di aggregazione HLL_COMBINE restituisce un tipo di dati HLLSKETCH che combina tutti i valori HLLSKETCH di input.

La combinazione di due o più HyperLogLog schizzi è un nuovo HLLSKETCH che incapsula informazioni sull'unione dei valori distinti rappresentati da ogni schizzo di input. Dopo aver combinato

gli schizzi, Amazon Redshift estrae la cardinalità dell'unione di due o più set di dati. Per ulteriori informazioni su come combinare più schizzi, consultare Esempio: restituisci uno HyperLogLog schizzo combinando più schizzi.

Sintassi

```
HLL_COMBINE (hllsketch_expression)
```

Argomento

hllsketch_expression

Qualsiasi espressione valida che restituisce un tipo HLLSKETCH, come ad esempio un nome di colonna. Il valore di input è il tipo di dati HLLSKETCH.

Tipo restituito

La funzione HLL_COMBINE restituisce un tipo HLLSKETCH.

Esempi

L'esempio seguente restituisce i valori HLLSKETCH combinati nella tabella `hll_table`.

```
CREATE TABLE a_table(an_int INT, b_int INT);
INSERT INTO a_table VALUES (1,1), (2,1), (3,1), (4,1), (1,2), (2,2), (3,2), (4,2),
 (5,2), (6,2);

CREATE TABLE hll_table (sketch HLLSKETCH);
INSERT INTO hll_table select hll_create_sketch(an_int) from a_table group by b_int;

SELECT hll_combine(sketch) AS sketches FROM hll_table;
sketches
--------------------------------------------------------------------------------------------------
{"version":1,"logm":15,"sparse":{"indices":
[20812342,20850007,22362299,40314817,42650774,47158030],"values":[1,2,1,3,2,1]}}
(1 row)
```

Funzione HLL_COMBINE_SKETCHES

HLL_COMBINE_SKETCHES è una funzione scalare che prende come input due valori HLLSKETCH e li combina in un singolo HLLSKETCH.

La combinazione di due o più HyperLogLog schizzi è un nuovo HLLSKETCH che incapsula informazioni sull'unione dei valori distinti rappresentati da ogni schizzo di input.

Sintassi

```
HLL_COMBINE_SKETCHES (hllsketch_expression1, hllsketch_expression2)
```

Argomento

hllsketch_expression1 e hllsketch_expression2

Qualsiasi espressione valida che restituisce un tipo HLLSKETCH, come ad esempio un nome di colonna.

Tipo restituito

La funzione HLL_COMBINE_SKETCHES restituisce un tipo HLLSKETCH.

Esempi

L'esempio seguente restituisce i valori HLLSKETCH combinati nella tabella `hll_table`.

```
WITH tbl1(x, y)
     AS (SELECT Hll_create_sketch(1),
                Hll_create_sketch(2)
         UNION ALL
         SELECT Hll_create_sketch(3),
                Hll_create_sketch(4)
         UNION ALL
         SELECT Hll_create_sketch(5),
                Hll_create_sketch(6)
         UNION ALL
         SELECT Hll_create_sketch(7),
                Hll_create_sketch(8)),
     tbl2(x, y)
     AS (SELECT Hll_create_sketch(9),
                Hll_create_sketch(10)
         UNION ALL
         SELECT Hll_create_sketch(11),
                Hll_create_sketch(12)
         UNION ALL
         SELECT Hll_create_sketch(13),
                Hll_create_sketch(14)
```

```
        UNION ALL
        SELECT Hll_create_sketch(15),
               Hll_create_sketch(16)
        UNION ALL
        SELECT Hll_create_sketch(NULL),
               Hll_create_sketch(NULL)),
     tbl3(x, y)
     AS (SELECT *
         FROM   tbl1
         UNION ALL
         SELECT *
         FROM   tbl2)
SELECT Hll_combine_sketches(x, y)
FROM   tbl3;
```

Funzioni JSON

Argomenti

- Funzione IS_VALID_JSON
- Funzione IS_VALID_JSON_ARRAY
- Funzione JSON_ARRAY_LENGTH
- Funzione JSON_EXTRACT_ARRAY_ELEMENT_TEXT
- Funzione JSON_EXTRACT_PATH_TEXT
- Funzione JSON_PARSE
- Funzione CAN_JSON_PARSE
- Funzione JSON_SERIALIZE
- Funzione JSON_SERIALIZE_TO_VARBYTE

Quando è necessario memorizzare un insieme relativamente piccolo di coppie chiave-valore, è possibile risparmiare spazio memorizzando i dati nel formato JSON. Poiché le stringhe JSON possono essere memorizzate in una singola colonna, l'utilizzo di JSON potrebbe essere più efficiente rispetto all'archiviazione dei dati in formato tabulare. Ad esempio, supponiamo di disporre di una tabella sparsa, in cui è necessario avere molte colonne per rappresentare a pieno tutti gli attributi possibili, ma la maggior parte dei valori delle colonne è NULL per ogni riga o colonna data. Usando JSON per l'archiviazione, si potrebbe essere in grado di memorizzare i dati per una riga in coppie chiave:valore in una singola stringa JSON ed eliminare le colonne della tabella scarsamente popolate.

Inoltre, è possibile modificare facilmente le stringhe JSON per memorizzare coppie chiavi:valore aggiuntive senza dover aggiungere colonne a una tabella.

È consigliabile usare un JSON con parsimonia. JSON non è una buona scelta per archiviare set di dati più grandi perché, archiviando dati disparati in una singola colonna, JSON non usa abitualmente l'architettura di archiviazione di colonne di Amazon Redshift. Sebbene Amazon Redshift supporti le funzioni JSON su colonne CHAR e VARCHAR, per l'elaborazione dei dati in formato di serializzazione JSON consigliamo di utilizzare SUPER. SUPER utilizza una rappresentazione senza schema post-analisi in grado di eseguire query in modo efficiente sui dati gerarchici. Per ulteriori informazioni sui tipi di dati SUPER, consultare Importazione e query di dati semistrutturati in Amazon Redshift.

JSON utilizza stringhe di testo con codifica UTF-8, pertanto le stringhe JSON possono essere memorizzate come tipi di dati CHAR o VARCHAR. Utilizzare VARCHAR se le stringhe includono caratteri multibyte.

Le stringhe JSON devono essere formattate in modo corretto con JSON, in base alle seguenti regole:

- Il JSON di livello radice può essere un oggetto JSON o un array JSON. Un oggetto JSON è un insieme non ordinato di coppie di chiave:valore separate da virgole racchiuse da parentesi graffe.

  Ad esempio, `{"one":1, "two":2}`
- Un array JSON è un insieme ordinato di valori separati da virgola racchiusi tra parentesi.

  Un esempio è quanto segue: `["first", {"one":1}, "second", 3, null]`
- Gli array JSON utilizzano un indice basato su zero; il primo elemento di un array è in posizione 0. In una coppia chiave:valore JSON, la chiave è una stringa racchiusa tra virgolette doppie.
- Un valore JSON può essere uno dei seguenti:
  - Oggetto JSON
  - Array JSON
  - stringa tra virgolette doppie
  - numero (intero e a virgola mobile)
  - booleano
  - null
- Gli oggetti vuoti e gli array vuoti sono valori JSON validi.
- I campi JSON fanno distinzione tra maiuscole e minuscole.

- Lo spazio bianco tra gli elementi strutturali JSON (ad esempio `{ }, [ ]`) viene ignorato.

Le funzioni JSON di Amazon Redshift e il comando COPY di Amazon Redshift usano gli stessi metodi per lavorare con i dati in formato JSON. Per ulteriori informazioni sull'utilizzo di JSON, consulta COPY dal formato JSON

Funzione IS_VALID_JSON

La funzione IS_VALID_JSON convalida una stringa JSON. La funzione restituisce un valore booleano di `true` se la stringa è in formato JSON corretto o `false` se la stringa è in formato errato. Per convalidare un array JSON, utilizzare Funzione IS_VALID_JSON_ARRAY

Per ulteriori informazioni, consultare Funzioni JSON.

Sintassi

```
IS_VALID_JSON('json_string')
```

Argomenti

json_string

Una stringa o espressione che valuta una stringa JSON.

Tipo restituito

BOOLEAN

Esempi

Per creare una tabella e inserire le stringhe JSON per i test, utilizza l'esempio seguente.

```
CREATE TABLE test_json(id int IDENTITY(0,1), json_strings VARCHAR);

-- Insert valid JSON strings --
INSERT INTO test_json(json_strings) VALUES
('{"a":2}'),
('{"a":{"b":{"c":1}}}'),
('{"a": [1,2,"b"]}');

-- Insert invalid JSON strings --
INSERT INTO test_json(json_strings) VALUES
```

```
('{{}}'),
('{1:"a"}'),
('[1,2,3]');
```

Per convalidare le stringhe nell'esempio precedente, utilizza l'esempio seguente.

```
SELECT id, json_strings, IS_VALID_JSON(json_strings)
FROM test_json
ORDER BY id;

+----+--------------------+---------------+
| id |    json_strings    | is_valid_json |
+----+--------------------+---------------+
|  0 | {"a":2}            | true          |
|  4 | {"a":{"b":{"c":1}}} | true          |
|  8 | {"a": [1,2,"b"]}   | true          |
| 12 | {{}}               | false         |
| 16 | {1:"a"}            | false         |
| 20 | [1,2,3]            | false         |
+----+--------------------+---------------+
```

Funzione IS_VALID_JSON_ARRAY

La funzione IS_VALID_JSON-ARRAY convalida un array JSON. La funzione restituisce un valore booleano `true` se l'array è in formato JSON corretto o `false` se l'array è in formato errato. Per convalidare una stringa JSON, utilizzare Funzione IS_VALID_JSON

Per ulteriori informazioni, consultare Funzioni JSON.

Sintassi

```
IS_VALID_JSON_ARRAY('json_array')
```

Argomenti

json_array

Una stringa o espressione che valuta un array JSON.

Tipo restituito

BOOLEAN

Esempi

Per creare una tabella e inserire le stringhe JSON per i test, utilizza l'esempio seguente.

```
CREATE TABLE test_json_arrays(id int IDENTITY(0,1), json_arrays VARCHAR);

-- Insert valid JSON array strings --
INSERT INTO test_json_arrays(json_arrays)
VALUES('[]'),
('["a","b"]'),
('["a",["b",1,["c",2,3,null]]]');

-- Insert invalid JSON array strings --
INSERT INTO test_json_arrays(json_arrays)
VALUES('{"a":1}'),
('a'),
('[1,2,]');
```

Per convalidare le stringhe nell'esempio precedente, utilizza l'esempio seguente.

```
SELECT json_arrays, IS_VALID_JSON_ARRAY(json_arrays)
FROM test_json_arrays ORDER BY id;

+-----------------------------+---------------------+
|         json_arrays         | is_valid_json_array |
+-----------------------------+---------------------+
| []                          | true                |
| ["a","b"]                   | true                |
| ["a",["b",1,["c",2,3,null]]] | true                |
| {"a":1}                     | false               |
| a                           | false               |
| [1,2,]                      | false               |
+-----------------------------+---------------------+
```

Funzione JSON_ARRAY_LENGTH

La funzione JSON_ARRAY_LENGTH restituisce il numero di elementi nell'array esterno di una stringa JSON. Se l'argomento null_if_invalid è impostato su `true` e la stringa JSON non è valida, la funzione restituisce NULL invece di restituire un errore.

Per ulteriori informazioni, consulta Funzioni JSON.

Sintassi

```
JSON_ARRAY_LENGTH('json_array' [, null_if_invalid ] )
```

Argomenti

json_array

Un array JSON correttamente formattato.

null_if_invalid

(Facoltativo) Un valore BOOLEAN che specifica se restituire NULL se la stringa JSON di input non è valida, invece di restituire un errore. Per restituire NULL se JSON non è valido, specifica `true` (`t`). Per restituire un errore se JSON non è valido, specificare `false` (`f`). Il valore predefinito è `false`.

Tipo restituito

INTEGER

Esempi

Per restituire il numero di elementi nell'array, utilizza l'esempio seguente.

```
SELECT JSON_ARRAY_LENGTH('[11,12,13,{"f1":21,"f2":[25,26]},14]');

+-------------------+
| json_array_length |
+-------------------+
|                 5 |
+-------------------+
```

Per restituire un errore poiché JSON non è valido, utilizza l'esempio seguente.

```
SELECT JSON_ARRAY_LENGTH('[11,12,13,{"f1":21,"f2":[25,26]},14');

ERROR: invalid json array object [11,12,13,{"f1":21,"f2":[25,26]},14
```

Per impostare null_if_invalid su true, in modo che l'istruzione restituisca NULL invece di restituire un errore di formato JSON non valido, utilizza l'esempio seguente.

```
SELECT JSON_ARRAY_LENGTH('[11,12,13,{"f1":21,"f2":[25,26]},14',true);

+------------------+
| json_array_length |
+------------------+
| NULL              |
+------------------+
```

Funzione JSON_EXTRACT_ARRAY_ELEMENT_TEXT

JSON_EXTRACT_ARRAY_ELEMENT_TEXT restituisce un elemento di array JSON nell'array più esterno di una stringa JSON, utilizzando un indice con base zero. Il primo elemento in un array è in posizione 0. Se l'indice è negativo o non vincolato, JSON_EXTRACT_ARRAY_ELEMENT_TEXT restituisce una stringa vuota. Se l'argomento null_if_invalid è impostato su `true` e la stringa JSON non è valida, la funzione restituisce `NULL` invece di restituire un errore.

Per ulteriori informazioni, consulta Funzioni JSON.

Sintassi

```
JSON_EXTRACT_ARRAY_ELEMENT_TEXT('json string', pos [, null_if_invalid ] )
```

Argomenti

json_string

Una stringa JSON correttamente formattata.

pos

Un `INTEGER` che rappresenta l'indice dell'elemento array da restituire, utilizzando un indice di array con base zero.

null_if_invalid

(Facoltativo) Un valore `BOOLEAN` che specifica se restituire `NULL` se la stringa JSON di input non è valida, invece di restituire un errore. Per restituire `NULL` se JSON non è valido, specifica `true` (`t`). Per restituire un errore se JSON non è valido, specificare `false` (`f`). Il valore predefinito è `false`.

Tipo restituito

VARCHAR

Una stringa VARCHAR che rappresenta l'elemento dell'array JSON a cui fa riferimento pos.

Esempi

Per restituire un elemento di array alla posizione 2, che è il terzo elemento di un indice di array a base zero, utilizza l'esempio seguente.

```
SELECT JSON_EXTRACT_ARRAY_ELEMENT_TEXT('[111,112,113]', 2);

+---------------------------------+
| json_extract_array_element_text |
+---------------------------------+
|                             113 |
+---------------------------------+
```

Per restituire un errore poiché JSON non è valido, utilizza l'esempio seguente.

```
SELECT JSON_EXTRACT_ARRAY_ELEMENT_TEXT('["a",["b",1,["c",2,3,null,]]]',1);

ERROR: invalid json array object ["a",["b",1,["c",2,3,null,]]]
```

Per impostare null_if_invalid su true, in modo che l'istruzione restituisca NULL invece di restituire un errore di formato JSON non valido, utilizza l'esempio seguente.

```
SELECT JSON_EXTRACT_ARRAY_ELEMENT_TEXT('["a",["b",1,["c",2,3,null,]]]',1,true);

+---------------------------------+
| json_extract_array_element_text |
+---------------------------------+
| NULL                            |
+---------------------------------+
```

Funzione JSON_EXTRACT_PATH_TEXT

La funzione JSON_EXTRACT_PATH_TEXT restituisce il valore per la coppia chiave-valore a cui fa riferimento una serie di elementi di percorso in una stringa JSON. Il percorso JSON può

essere nidificato fino a cinque livelli di profondità. Gli elementi del percorso fanno distinzione tra maiuscole e minuscole. Se un elemento del percorso non esiste nella stringa JSON, JSON_EXTRACT_PATH_TEXT restituisce NULL.

Se l'argomento null_if_invalid è impostato su `true` e la stringa JSON non è valida, la funzione restituisce NULL invece di restituire un errore.

Per informazioni sulle funzioni JSON aggiuntive, consulta Funzioni JSON. Per ulteriori informazioni sull'utilizzo di JSON, consulta COPY dal formato JSON.

Sintassi

```
JSON_EXTRACT_PATH_TEXT('json_string', 'path_elem' [,'path_elem'[, …] ]
 [, null_if_invalid ] )
```

Argomenti

json_string

Una stringa JSON correttamente formattata.

path_elem

Un elemento di percorso in una stringa JSON. Un elemento di percorso è obbligatorio. È possibile specificare elementi aggiuntivi del percorso, fino a cinque livelli di profondità.

null_if_invalid

(Facoltativo) Un valore BOOLEAN che specifica se restituire NULL se la stringa JSON di input non è valida, invece di restituire un errore. Per restituire NULL se JSON non è valido, specifica `true` (`t`). Per restituire un errore se JSON non è valido, specificare `false` (`f`). Il valore predefinito è `false`.

In una stringa JSON, Amazon Redshift riconosce \n come carattere newline e `\t` come carattere di tabulazione. Per caricare una barra rovesciata, crea una sequenza di escape con una barra rovesciata (\\). Per ulteriori informazioni, consultare Caratteri escape in JSON.

Tipo restituito

VARCHAR

Una stringa VARCHAR che rappresenta il valore JSON cui fanno riferimento gli elementi di percorso.

Esempi

Per restituire il valore per il percorso 'f4', 'f6', utilizza l'esempio seguente.

```
SELECT JSON_EXTRACT_PATH_TEXT('{"f2":{"f3":1},"f4":{"f5":99,"f6":"star"}}','f4', 'f6');

+------------------------+
| json_extract_path_text |
+------------------------+
| star                   |
+------------------------+
```

Per restituire un errore poiché JSON non è valido, utilizza l'esempio seguente.

```
SELECT JSON_EXTRACT_PATH_TEXT('{"f2":{"f3":1},"f4":{"f5":99,"f6":"star"}','f4', 'f6');

ERROR: invalid json object {"f2":{"f3":1},"f4":{"f5":99,"f6":"star"}
```

Per impostare null_if_invalid su true, in modo che l'istruzione restituisca NULL per il formato JSON non valido invece di restituire un errore, utilizza l'esempio seguente.

```
SELECT JSON_EXTRACT_PATH_TEXT('{"f2":{"f3":1},"f4":{"f5":99,"f6":"star"}','f4',
 'f6',true);

+------------------------+
| json_extract_path_text |
+------------------------+
| NULL                   |
+------------------------+
```

Per restituire il valore per il percorso.'farm', 'barn', 'color', dove il valore recuperato si trova al terzo livello, utilizza l'esempio seguente. Questo esempio è formattato con uno strumento JSON Lint per semplificarne la lettura.

```
SELECT JSON_EXTRACT_PATH_TEXT('{
    "farm": {
        "barn": {
            "color": "red",
            "feed stocked": true
        }
    }
}', 'farm', 'barn', 'color');
+------------------------+
| json_extract_path_text |
+------------------------+
| red                    |
+------------------------+
```

Per restituire NULL perché l'elemento `'color'` risulta mancante, utilizza l'esempio seguente. Questo esempio è formattato con uno strumento JSON Lint.

```
SELECT JSON_EXTRACT_PATH_TEXT('{
    "farm": {
        "barn": {}
    }
}', 'farm', 'barn', 'color');

+------------------------+
| json_extract_path_text |
+------------------------+
| NULL                   |
+------------------------+
```

Se il formato JSON è valido, il tentativo di estrarre un elemento mancante restituisce NULL.

Per restituire il valore per il percorso `'house', 'appliances', 'washing machine', 'brand'`, utilizza l'esempio seguente.

```
SELECT JSON_EXTRACT_PATH_TEXT('{
  "house": {
    "address": {
      "street": "123 Any St.",
      "city": "Any Town",
      "state": "FL",
      "zip": "32830"
    },
```

```
    "bathroom": {
      "color": "green",
      "shower": true
    },
    "appliances": {
      "washing machine": {
        "brand": "Any Brand",
        "color": "beige"
      },
      "dryer": {
        "brand": "Any Brand",
        "color": "white"
      }
    }
  }
}', 'house', 'appliances', 'washing machine', 'brand');

+------------------------+
| json_extract_path_text |
+------------------------+
| Any Brand              |
+------------------------+
```

L'esempio seguente crea una tabella di esempio e la popola con valori SUPER, quindi restituisce il valore del percorso per entrambe le righe. `'f2'`

```
CREATE TABLE json_example(id INT, json_text SUPER);

INSERT INTO json_example VALUES
(1, JSON_PARSE('{"f2":{"f3":1},"f4":{"f5":99,"f6":"star"}}')),
(2, JSON_PARSE('{
    "farm": {
        "barn": {
            "color": "red",
            "feed stocked": true
        }
    }
}'));

SELECT * FROM json_example;
id          | json_text
------------+-------------------------------------------
1           | {"f2":{"f3":1},"f4":{"f5":99,"f6":"star"}}
```

```
2           | {"farm":{"barn":{"color":"red","feed stocked":true}}}


SELECT id, JSON_EXTRACT_PATH_TEXT(JSON_SERIALIZE(json_text), 'f2') FROM json_example;

id          | json_text
------------+-------------------------------------------
1           | {"f3":1}
2           |
```

Funzione JSON_PARSE

La funzione JSON_PARSE analizza i dati in formato JSON e li converte nella rappresentazione SUPER.

Per importare nel tipo di dati SUPER utilizzando il comando INSERT o UPDATE, utilizza la funzione JSON_PARSE. Quando utilizzi JSON_PARSE() per analizzare le stringhe JSON in valori SUPER, si applicano alcune restrizioni. Per ulteriori informazioni, consulta Opzioni di analisi per SUPER.

Sintassi

```
JSON_PARSE( {json_string | binary_value} )
```

Argomenti

json_string

Un'espressione che restituisce JSON serializzato come tipo VARBYTE o VARCHAR.

binary_value

Un valore binario di tipo VARBYTE.

Tipo restituito

SUPER

Esempi

Per convertire l'array JSON `[10001,10002,"abc"]` nel tipo di dati SUPER, utilizza l'esempio seguente.

```
SELECT JSON_PARSE('[10001,10002,"abc"]');

+---------------------+
|     json_parse      |
+---------------------+
| [10001,10002,"abc"] |
+---------------------+
```

Per assicurare che la funzione abbia convertito l'array JSON nel tipo di dati SUPER, utilizza l'esempio seguente. Per ulteriori informazioni, consulta Funzione JSON_TYPEOF

```
SELECT JSON_TYPEOF(JSON_PARSE('[10001,10002,"abc"]'));

+-------------+
| json_typeof |
+-------------+
| array       |
+-------------+
```

Funzione CAN_JSON_PARSE

La funzione CAN_JSON_PARSE analizza i dati in formato JSON e restituisce `true` se il risultato può essere convertito in un valore SUPER utilizzando le funzione JSON_PARSE.

Sintassi

```
CAN_JSON_PARSE( {json_string | binary_value} )
```

Argomenti

json_string

Un'espressione che restituisce JSON serializzato nel formato VARBYTE o VARCHAR.

binary_value

Un valore binario di tipo VARBYTE.

Tipo restituito

BOOLEAN

Esempi

Per vedere se è possibile convertire l'array JSON `[10001,10002,"abc"]` nel tipo di dati SUPER, utilizza l'esempio seguente.

```
SELECT CAN_JSON_PARSE('[10001,10002,"abc"]');

+----------------+
| can_json_parse |
+----------------+
| true           |
+----------------+
```

Funzione JSON_SERIALIZE

La funzione JSON_SERIALIZE serializza un'espressione SUPER nella rappresentazione testuale JSON per seguire RFC 8259. Per ulteriori informazioni su tale RFC, vedere The JavaScript Object Notation (JSON) Data Interchange Format.

Il limite di dimensione SUPER è approssimativamente uguale al limite di blocco e il limite VARCHAR è più piccolo del limite di dimensione SUPER. Pertanto, quando il formato JSON supera il limite varchar del sistema la funzione JSON_SERIALIZE restituisce un errore. Per verificare la dimensione di un'espressione SUPER, consulta la funzione JSON_SIZE.

Sintassi

```
JSON_SERIALIZE(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

VARCHAR

Esempi

Per serializzare un valore SUPER in una stringa, utilizza l'esempio seguente.

```
SELECT JSON_SERIALIZE(JSON_PARSE('[10001,10002,"abc"]'));

+---------------------+
|   json_serialize    |
+---------------------+
| [10001,10002,"abc"] |
+---------------------+
```

Funzione JSON_SERIALIZE_TO_VARBYTE

La funzione JSON_SERIALIZE_TO_VARBYTE converte un valore SUPER in una stringa JSON simile a JSON_SERIALIZE(), ma archiviata invece in un valore VARBYTE.

Sintassi

```
JSON_SERIALIZE_TO_VARBYTE(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

VARBYTE

Esempi

Per serializzare un valore SUPER e restituire il risultato in formato VARBYTE, utilizza l'esempio seguente.

```
SELECT JSON_SERIALIZE_TO_VARBYTE(JSON_PARSE('[10001,10002,"abc"]'));

+----------------------------------------+
|       json_serialize_to_varbyte        |
+----------------------------------------+
| 5b31303030312c31303030322c22616263225d |
+----------------------------------------+
```

Per serializzare un valore SUPER e convertire il risultato in formato VARCHAR, utilizza l'esempio seguente. Per ulteriori informazioni, consulta Funzione CAST.

```
SELECT CAST((JSON_SERIALIZE_TO_VARBYTE(JSON_PARSE('[10001,10002,"abc"]'))) AS VARCHAR);

+---------------------------+
| json_serialize_to_varbyte |
+---------------------------+
| [10001,10002,"abc"]       |
+---------------------------+
```

Funzioni di machine learning

Grazie ad Amazon Redshift machine learning (ML), è possibile addestrare modelli di ML utilizzando istruzioni SQL e richiamarli nelle query SQL per la previsione. La spiegabilità del modello Amazon Redshift include valori di importanza per le funzionalità per aiutarti a capire in che modo ogni attributo nei tuoi dati di formazione contribuisce al risultato previsto.

Di seguito vengono fornite le descrizioni delle funzioni machine learning per SQL supportate da Amazon Redshift.

Argomenti

- funzione EXPLAIN_MODEL

funzione EXPLAIN_MODEL

La funzione EXPLAIN_MODEL restituisce un tipo di dati SUPER che contiene un report di spiegabilità del modello in formato JSON. Il report di spiegabilità contiene informazioni sul valore Shapley per tutte le caratteristiche del modello.

La funzione EXPLAIN_MODEL attualmente supporta solo i modelli AUTO ON o AUTO OFF XGBoost.

Quando il report di spiegabilità non è disponibile, la funzione restituisce gli stati visualizzati sullo stato di avanzamento del modello. Tra queste vi sono `Waiting for training job to complete`, `Waiting for processing job to complete` e `Processing job failed`.

Quando si esegue l'istruzione CREATE MODEL, lo stato di spiegazione diventa `Waiting for training job to complete`. Quando il modello è stato addestrato e viene inviata una richiesta di spiegazione, lo stato di spiegazione diventa `Waiting for processing job to complete`.

Quando la spiegazione del modello viene completata correttamente, è disponibile il report completo di spiegabilità. In caso contrario, lo stato diventa `Processing job failed`.

Quando si esegue l'istruzione CREATE MODEL, è possibile utilizzare il parametro opzionale `MAX_RUNTIME` per specificare la quantità massima di tempo che l'addestramento deve richiedere. Quando la creazione del modello raggiunge tale periodo di tempo, Amazon Redshift interrompe la creazione del modello. Se raggiungi tale limite di tempo durante la creazione di un modello di autopilota, Amazon Redshift restituirà il modello migliore finora. La spiegabilità del modello diventa disponibile una volta terminato l'addestramento sul modello, quindi se `MAX_RUNTIME` è impostato su un periodo di tempo basso, il rapporto di spiegabilità potrebbe non essere disponibile. Il tempo di addestramento varia e dipende dalla complessità del modello, dalla dimensione dei dati e da altri fattori.

Sintassi

```
EXPLAIN_MODEL ('schema_name.model_name')
```

Argomento

schema_name

Il nome dello schema. Se non viene specificato alcun nome_schema, viene selezionato lo schema corrente.

model_name

Il nome del modello. Il nome del modello in uno schema deve essere unico.

Tipo restituito

La funzione EXPLAIN_MODEL restituisce un tipo di dati SUPER, come illustrato di seguito.

```
{"version":"1.0","explanations":{"kernel_shap":{"label0":{"global_shap_values":
{"x0":0.05,"x1":0.10,"x2":0.30,"x3":0.15},"expected_value":0.50}}}}
```

Esempi

L'esempio seguente restituisce lo stato di spiegazione `waiting for training job to complete`.

```
select explain_model('customer_churn_auto_model');
                 explain_model
-------------------------------------------------------
{"explanations":"waiting for training job to complete"}
(1 row)
```

Quando la spiegazione del modello viene completata correttamente, il report completo di spiegabilità è disponibile come segue.

```
select explain_model('customer_churn_auto_model');
                                       explain_model
-------------------------------------------------------------------------------------------------
{"version":"1.0","explanations":{"kernel_shap":{"label0":{"global_shap_values":
{"x0":0.05386043365892927,"x1":0.10801289723274592,"x2":0.23227865827017378,"x3":0.067668513394
(1 row)
```

Poiché la funzione EXPLAIN_MODEL restituisce il tipo di dati SUPER, è possibile eseguire una query sul rapporto di spiegabilità. Facendo questo, è possibile estrarre `global_shap_values`, `expected_value` o valori Shapley specifici della funzione.

Il seguente esempio estrae `global_shap_values` per il modello.

```
select json_table.report.explanations.kernel_shap.label0.global_shap_values from
 (select explain_model('customer_churn_auto_model') as report) as json_table;
                                                    global_shap_values
-------------------------------------------------------------------------------------------------
{"state":0.10983770427197151,"account_length":0.1772441398408543,"area_code":0.0862682396863959
(1 row)
```

Il seguente esempio estrae `global_shap_values` per la funzione x0.

```
select json_table.report.explanations.kernel_shap.label0.global_shap_values.x0 from
 (select explain_model('customer_churn_auto_model') as report) as json_table;
          x0
------------------------
  0.05386043365892927
(1 row)
```

Se il modello viene creato in uno schema specifico e si ha accesso al modello creato, è possibile eseguire una query sulla spiegazione del modello come illustrato di seguito.

```
-- Check the current schema
SHOW search_path;
   search_path
-----------------
  $user, public
(1 row)
-- If you have the privilege to access the model explanation
-- in `test_schema`
SELECT explain_model('test_schema.test_model_name');
                       explain_model
-------------------------------------------------------------
{"explanations":"waiting for training job to complete"}
(1 row)
```

Funzioni matematiche

Argomenti

- Funzione LOG
- Funzione MOD
- Funzione PI
- Funzione POWER
- Funzioni RADIANS
- Funzione RANDOM
- Funzione ROUND
- Funzione SIN
- Funzione SIGN
- Funzione SQRT
- Funzione TAN
- Funzione TRUNC

In questa sezione sono descritti funzioni e operatori matematici supportati in Amazon Redshift.

Simboli degli operatori matematici

La tabella seguente elenca gli operatori matematici supportati.

Operatori supportati

Operatore	Descrizione	Esempio	Risultato
+	addizione	2 + 3	5
-	sottrazione	2 - 3	-1
*	moltiplicazione	2 * 3	6
/	divisione	4 / 2	2
%	modulo	5 % 4	1
^	potenza	2,0 ^ 3,0	8

Operatore	Descrizione	Esempio	Risultato
|/	radice quadrata	| / 25,0	5
||/	radice cubica	|| / 27,0	3
@	valore assoluto	@ -5,0	5
<<	spostamento bit a bit a sinistra	1 << 4	16
>>	spostamento bit a bit a destra	8 >> 2	2
&	bitwise e	8 & 2	0

Esempi

Gli esempi seguenti utilizzano il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per calcolare la commissione pagata più una gestione di 2,00 USD per una determinata transazione, utilizza l'esempio seguente.

```
SELECT
    commission,
    (commission + 2.00) AS comm
FROM
    sales
WHERE
    salesid = 10000;

+------------+-------+
| commission | comm  |
+------------+-------+
```

```
|      28.05 | 30.05 |
+------------+-------+
```

Per calcolare il 20 percento del prezzo di vendita per una determinata transazione, utilizza l'esempio seguente.

```
SELECT pricepaid, (pricepaid * .20) as twentypct
FROM sales
WHERE salesid=10000;

+-----------+-----------+
| pricepaid | twentypct |
+-----------+-----------+
|       187 |      37.4 |
+-----------+-----------+
```

Per prevedere le vendite di biglietti in base a un modello di crescita continua, utilizza l'esempio seguente. In questo esempio, la sottoquery restituisce il numero di biglietti venduti nel 2008. Questo risultato è moltiplicato in modo esponenziale per un tasso di crescita continuo del 5% su 10 anni.

```
SELECT (SELECT SUM(qtysold) FROM sales, date
WHERE sales.dateid=date.dateid AND year=2008)^((5::float/100)*10) AS qty10years;

+------------------+
|    qty10years    |
+------------------+
| 587.664019657491 |
+------------------+
```

Per trovare il prezzo totale pagato e le commissioni per le vendite con un ID data che è maggiore o uguale a 2000, utilizza l'esempio seguente. Quindi sottrarre la commissione totale dal prezzo totale pagato.

```
SELECT SUM(pricepaid) AS sum_price, dateid,
SUM(commission) AS sum_comm, (SUM(pricepaid) - SUM(commission)) AS value
FROM sales
WHERE dateid >= 2000
GROUP BY dateid
ORDER BY dateid
LIMIT 10;
```

```
+-----------+--------+----------+-----------+
| sum_price | dateid | sum_comm |   value   |
+-----------+--------+----------+-----------+
|    305885 |   2000 | 45882.75 | 260002.25 |
|    316037 |   2001 | 47405.55 | 268631.45 |
|    358571 |   2002 | 53785.65 | 304785.35 |
|    366033 |   2003 | 54904.95 | 311128.05 |
|    307592 |   2004 |  46138.8 |  261453.2 |
|    333484 |   2005 |  50022.6 |  283461.4 |
|    317670 |   2006 |  47650.5 |  270019.5 |
|    351031 |   2007 | 52654.65 | 298376.35 |
|    313359 |   2008 | 47003.85 | 266355.15 |
|    323675 |   2009 | 48551.25 | 275123.75 |
+-----------+--------+----------+-----------+
```

Funzione ABS

ABS calcola il valore assoluto di un numero, in cui quel numero può essere un valore letterale o un'espressione che valuta un numero.

Sintassi

```
ABS(number)
```

Argomenti

numero

Numero o espressione che valuta un numero. Può essere anche di tipo `SMALLINT`, `INTEGER`, `BIGINT`, `DECIMAL`, `FLOAT4`, `FLOAT8` o `SUPER`.

Tipo restituito

ABS Restituisce lo stesso tipo di dati del suo argomento.

Esempi

Per calcolare il valore assoluto di `-38`, utilizza l'esempio seguente.

```
SELECT ABS(-38);

+-----+
```

```
| abs |
+-----+
|  38 |
+-----+
```

Per calcolare il valore assoluto di `(14-76)`, utilizza l'esempio seguente.

```
SELECT ABS(14-76);

+-----+
| abs |
+-----+
|  62 |
+-----+
```

Funzione ACOS

ACOS è una funzione trigonometrica che restituisce l'arco coseno di un numero. Il valore restituito è in radianti ed è compreso tra `0` e `PI`.

Sintassi

```
ACOS(number)
```

Argomenti

numero

Il parametro di input è un numero `DOUBLE PRECISION`.

Tipo restituito

`DOUBLE PRECISION`

Esempi

Per restituire l'arco coseno di `-1`, utilizza l'esempio seguente.

```
SELECT ACOS(-1);

+-------------------+
|       acos        |
```

```
+-------------------+
| 3.141592653589793 |
+-------------------+
```

Per convertire l'arcocoseno di `.5` al numero equivalente di gradi, utilizza l'esempio seguente.

```
SELECT (ACOS(.5) * 180/(SELECT PI())) AS degrees;

+-------------------+
|      degrees      |
+-------------------+
| 60.00000000000001 |
+-------------------+
```

Funzione ASIN

ASIN è una funzione trigonometrica che restituisce l'arco seno di un numero. Il valore restituito è in radianti ed è compreso tra `PI/2` e `-PI/2`.

Sintassi

```
ASIN(number)
```

Argomenti

numero

Il parametro di input è un numero `DOUBLE PRECISION`.

Tipo restituito

`DOUBLE PRECISION`

Esempi

Per restituire l'arco seno di 1, utilizza l'esempio seguente.

```
SELECT ASIN(1) AS halfpi;

+--------------------+
|       halfpi       |
+--------------------+
```

```
| 1.5707963267948966 |
+--------------------+
```

Per convertire l'arco seno di `.5` al numero equivalente di gradi, utilizza l'esempio seguente.

```
SELECT (ASIN(.5) * 180/(SELECT PI())) AS degrees;

+--------------------+
|      degrees       |
+--------------------+
| 30.000000000000004 |
+--------------------+
```

Funzione ATAN

ATAN è una funzione trigonometrica che restituisce l'arco tangente di un numero. Il valore restituito è in radianti ed è compreso tra `-PI` e `PI`.

Sintassi

```
ATAN(number)
```

Argomenti

numero

Il parametro di input è un numero `DOUBLE PRECISION`.

Tipo restituito

`DOUBLE PRECISION`

Esempi

Per restituire l'arco tangente di 1 e moltiplicarlo per 4, utilizza l'esempio seguente.

```
SELECT ATAN(1) * 4 AS pi;

+-------------------+
|        pi         |
+-------------------+
| 3.141592653589793 |
```

```
+-------------------+
```

Per convertire l'arco tangente di 1 al numero equivalente di gradi, utilizza l'esempio seguente.

```
SELECT (ATAN(1) * 180/(SELECT PI())) AS degrees;

+---------+
| degrees |
+---------+
|      45 |
+---------+
```

Funzione ATAN2

ATAN2 è una funzione trigonometrica che restituisce l'arco tangente di un numero diviso per un altro numero. Il valore restituito è in radianti ed è compreso tra `PI/2` e `-PI/2`.

Sintassi

```
ATAN2(number1, number2)
```

Argomenti

number1

Un numero `DOUBLE PRECISION`.

number2

Un numero `DOUBLE PRECISION`.

Tipo restituito

`DOUBLE PRECISION`

Esempi

Per restituire l'arco tangente di `2/2` e moltiplicarlo per 4, utilizza l'esempio seguente.

```
SELECT ATAN2(2,2) * 4 AS PI;

+-------------------+
```

```
|        pi         |
+-------------------+
| 3.141592653589793 |
+-------------------+
```

Per convertire l'arco tangente di `1/0` (che restituisce 0) al numero equivalente di gradi, utilizza l'esempio seguente.

```
SELECT (ATAN2(1,0) * 180/(SELECT PI())) AS degrees;

+---------+
| degrees |
+---------+
|      90 |
+---------+
```

Funzione CBRT

La funzione CBRT è una funzione matematica che calcola la radice cubica di un determinato numero.

Sintassi

```
CBRT(number)
```

Argomenti

CBRT accetta come argomento un numero `DOUBLE PRECISION`.

Tipo restituito

`DOUBLE PRECISION`

Esempi

L'esempio seguente utilizza il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per calcolare la radice cubica della commissione pagata per una determinata transazione, utilizza l'esempio seguente.

```
SELECT CBRT(commission) FROM sales WHERE salesid=10000;

+--------------------+
```

```
|        cbrt        |
+--------------------+
| 3.0383953904884344 |
+--------------------+
```

Funzione CEILING (oppure CEIL)

La funzione CEILING o CEIL viene utilizzata per arrotondare un numero fino al numero intero successivo. (L Funzione FLOOR arrotonda un numero fino al numero intero successivo.)

Sintassi

```
{CEIL | CEILING}(number)
```

Argomenti

numero

Il numero o l'espressione che restituisce un numero. Può essere anche di tipo SMALLINT, INTEGER, BIGINT, DECIMAL, FLOAT4, FLOAT8 o SUPER.

Tipo restituito

CEILING e CEIL restituiscono lo stesso tipo di dati come argomento.

Quando l'input è di tipo SUPER, l'output mantiene lo stesso tipo dinamico dell'input mentre il tipo statico rimane il tipo SUPER. Quando il tipo dinamico di SUPER non è un numero, Amazon Redshift restituisce un valore null.

Esempi

L'esempio seguente utilizza il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per calcolare il tetto della commissione pagata per una determinata transazione di vendita, utilizza l'esempio seguente.

```
SELECT CEILING(commission) FROM sales
WHERE salesid=10000;

+---------+
| ceiling |
```

```
+---------+
|      29 |
+---------+
```

Funzione COS

COS è una funzione trigonometrica che restituisce il coseno di un numero. Il valore restituito è in radianti ed è compreso tra `-1` e `1`, inclusi.

Sintassi

```
COS(double_precision)
```

Argomenti

numero

Il parametro di input è un numero `DOUBLE PRECISION`.

Tipo restituito

La funzione COS restituisce un numero `DOUBLE PRECISION`.

Esempi

Per restituire il coseno di `0`, utilizza l'esempio seguente.

```
SELECT COS(0);

+-----+
| cos |
+-----+
|   1 |
+-----+
```

Per restituire il coseno di `pi`, utilizza l'esempio seguente.

```
SELECT COS(PI());

+-----+
| cos |
+-----+
```

```
|  -1 |
+-----+
```

Funzione COT

COT è una funzione trigonometrica che restituisce la cotangente di un numero. Il parametro di input deve essere diverso da zero.

Sintassi

```
COT(number)
```

Argomento

numero

Il parametro di input è un numero DOUBLE PRECISION.

Tipo restituito

DOUBLE PRECISION

Esempi

Per restituire la cotangente di 1, utilizza l'esempio seguente.

```
SELECT COT(1);

+--------------------+
|        cot         |
+--------------------+
| 0.6420926159343306 |
+--------------------+
```

Funzione DEGREES

Converte un angolo in radianti nel suo equivalente in gradi.

Sintassi

```
DEGREES(number)
```

Argomento

numero

Il parametro di input è un numero `DOUBLE PRECISION`.

Tipo restituito

`DOUBLE PRECISION`

Esempi

Per restituire l'equivalente in gradi di 0,5 radianti, utilizza l'esempio seguente.

```
SELECT DEGREES(.5);

+-------------------+
|      degrees      |
+-------------------+
| 28.64788975654116 |
+-------------------+
```

Per convertire i radianti PI in gradi, utilizza l'esempio seguente.

```
SELECT DEGREES(pi());

+---------+
| degrees |
+---------+
|     180 |
+---------+
```

Funzione DEXP

La funzione DEXP restituisce il valore esponenziale nella notazione scientifica per un numero a precisione doppia. L'unica differenza tra le funzioni DEXP ed EXP è che il parametro per DEXP deve essere `DOUBLE PRECISION`.

Sintassi

```
DEXP(number)
```

Argomento

numero

Il parametro di input è un numero `DOUBLE PRECISION`.

Tipo restituito

`DOUBLE PRECISION`

Esempio

L'esempio seguente utilizza il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Utilizzare la funzione DEXP per prevedere le vendite di biglietti in base a un modello di crescita continua. In questo esempio, la sottoquery restituisce il numero di biglietti venduti nel 2008. Questo risultato è moltiplicato per il risultato della funzione DEXP, che specifica un tasso di crescita continua del 7% nel corso di 10 anni.

```
SELECT (SELECT SUM(qtysold)
FROM sales, date
WHERE sales.dateid=date.dateid
AND year=2008) * DEXP((7::FLOAT/100)*10) qty2010;

+-------------------+
|      qty2010      |
+-------------------+
| 695447.4837722216 |
+-------------------+
```

Funzione DLOG1

La funzione DLOG1 restituisce il logaritmo naturale del parametro di input. Sinonimo di Funzione LN.

Funzione DLOG10

La DLOG10 restituisce il logaritmo di base 10 del parametro di input.

Sinonimo di Funzione LOG.

Sintassi

```
DLOG10(number)
```

Argomento

numero

Il parametro di input è un numero DOUBLE PRECISION.

Tipo restituito

DOUBLE PRECISION

Esempio

Per restituire il logaritmo in base 10 del numero 100, utilizza l'esempio seguente.

```
SELECT DLOG10(100);

+--------+
| dlog10 |
+--------+
|      2 |
+--------+
```

Funzione EXP

La funzione EXP implementa la funzione esponenziale di un'espressione numerica o la base del logaritmo naturale, e, elevato alla potenza dell'espressione. La funzione EXP è l'inverso di Funzione LN.

Sintassi

```
EXP(expression)
```

Argomento

espressione

L'espressione di input deve essere un tipo di dati INTEGER, DECIMAL o DOUBLE PRECISION.

Tipo restituito

`DOUBLE PRECISION`

Esempio

L'esempio seguente utilizza il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Utilizzare la funzione EXP per prevedere le vendite di biglietti in base a un modello di crescita continua. In questo esempio, la sottoquery restituisce il numero di biglietti venduti nel 2008. Questo risultato è moltiplicato per il risultato della funzione EXP, che specifica un tasso di crescita continua del 7% nel corso di 10 anni.

```
SELECT (SELECT SUM(qtysold)
FROM sales, date
WHERE sales.dateid=date.dateid
AND year=2008) * EXP((7::FLOAT/100)*10) qty2018;

+-------------------+
|      qty2018      |
+-------------------+
| 695447.4837722216 |
+-------------------+
```

Funzione FLOOR

La funzione FLOOR arrotonda un numero fino al numero intero successivo.

Sintassi

```
FLOOR(number)
```

Argomento

numero

Il numero o l'espressione che restituisce un numero. Può essere anche di tipo `SMALLINT`, `INTEGER`, `BIGINT`, `DECIMAL`, `FLOAT4`, `FLOAT8` o `SUPER`.

Tipo restituito

FLOOR restituisce lo stesso tipo di dati del suo argomento.

Quando l'input è di tipo SUPER, l'output mantiene lo stesso tipo dinamico dell'input mentre il tipo statico rimane il tipo SUPER. Quando il tipo dinamico SUPER non è un numero, Amazon Redshift restituisce NULL.

Esempi

Gli esempi seguenti utilizzano il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per mostrare il valore della commissione pagata per una determinata transazione di vendita prima e dopo l'utilizzo della funzione FLOOR, utilizza l'esempio seguente.

```
SELECT commission
FROM sales
WHERE salesid=10000;

+------------+
| commission |
+------------+
|      28.05 |
+------------+

SELECT FLOOR(commission)
FROM sales
WHERE salesid=10000;

+-------+
| floor |
+-------+
|    28 |
+-------+
```

Funzione LN

Restituisce il logaritmo naturale del parametro di input.

Sinonimo di Funzione DLOG1.

Sintassi

```
LN(expression)
```

Argomento

espressione

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

> **Note**
>
> Questa funzione restituisce un errore per alcuni tipi di dati se l'espressione fa riferimento a una tabella creata dall'utente o una tabella di sistema STL o STV di Amazon Redshift.

Le espressioni con i seguenti tipi di dati generano un errore se fanno riferimento a una tabella creata dall'utente o di sistema. Le espressioni con questi tipi di dati vengono eseguite esclusivamente sul nodo principale:

- `BOOLEAN`
- `CHAR`
- `DATE`
- `DECIMAL` o `NUMERIC`
- `TIMESTAMP`
- `VARCHAR`

Le espressioni con i seguenti tipi di dati vengono eseguite correttamente su tabelle create dall'utente e su tabelle di sistema STL o STV:

- `BIGINT`
- `DOUBLE PRECISION`
- `INTEGER`
- `REAL`
- `SMALLINT`

Tipo restituito

La funzione LN restituisce lo stesso tipo dell'input expression.

Esempi

Per restituire il logaritmo naturale o il logaritmo in base e del numero 2,718281828, utilizza l'esempio seguente.

```
SELECT LN(2.718281828);

+--------------------+
|         ln         |
+--------------------+
| 0.9999999998311267 |
+--------------------+
```

Si noti che la risposta è quasi uguale a 1.

L'esempio seguente utilizza il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire il logaritmo naturale dei valori nella colonna userid nella tabella USERS, utilizza l'esempio seguente.

```
SELECT username, LN(userid) FROM users ORDER BY userid LIMIT 10;

+----------+--------------------+
| username |         ln         |
+----------+--------------------+
| JSG99FHE |                  0 |
| PGL08LJI | 0.6931471805599453 |
| IFT66TXU | 1.0986122886681098 |
| XDZ38RDD | 1.3862943611198906 |
| AEB55QTM | 1.6094379124341003 |
| NDQ15VBM |  1.791759469228055 |
| OWY35QYB | 1.9459101490553132 |
| AZG78YIP | 2.0794415416798357 |
| MSD36KVR | 2.1972245773362196 |
| WKW41AIW |  2.302585092994046 |
+----------+--------------------+
```

Funzione LOG

Restituisce il logaritmo di un numero.

Se usi questa funzione per calcolare il logaritmo in base 10, puoi anche usare Funzione DLOG10.

Sintassi

```
LOG([base, ]argument)
```

Parametri

base

(Facoltativo) La base della funzione logaritmo. Questo numero deve essere positivo e non può essere uguale a 1. Se questo parametro viene omesso, Amazon Redshift calcola il logaritmo in base 10 di argument.

argument

L'argomento della funzione logaritmica. Questo numero deve essere positivo. Se il valore di argument è 1, la funzione restituisce 0.

Tipo restituito

La funzione LOG restituisce un numero DOUBLE PRECISION.

Esempi

Per trovare il logaritmo in base 2 del numero 100, utilizza l'esempio seguente.

```
SELECT LOG(2, 100);
+-------------------+
|        log        |
+-------------------+
| 6.643856189774725 |
+-------------------+
```

Per trovare il logaritmo in base 10 del numero 100, utilizza l'esempio seguente. Tieni presente che se ometti il parametro base, Amazon Redshift presuppone una base di 10.

```
SELECT LOG(100);
```

```
+-----+
| log |
+-----+
|   2 |
+-----+
```

Funzione MOD

Restituisce il resto di due numeri, altrimenti nota come operazione modulo. Per calcolare il risultato, il primo parametro viene diviso per il secondo.

Sintassi

```
MOD(number1, number2)
```

Argomenti

number1

Il primo parametro di input è un numero `INTEGER`, `SMALLINT`, `BIGINT` o `DECIMAL`. Se uno dei parametri è di tipo `DECIMAL`, anche l'altro parametro deve essere di tipo `DECIMAL`. Se uno dei parametri è un valore `INTEGER`, l'altro parametro può essere `INTEGER`, `SMALLINT` o `BIGINT`. Entrambi i parametri possono essere anche `SMALLINT` o `BIGINT`, ma un parametro non può essere `SMALLINT` se l'altro è `BIGINT`.

number2

Il secondo parametro è un numero `INTEGER`, `SMALLINT`, `BIGINT` o `DECIMAL`. Le stesse regole sui tipi di dati si applicano a number2 così come a number1.

Tipo restituito

Il tipo di restituzione della funzione MOD è lo stesso tipo numerico dei parametri di input, se entrambi i parametri di input sono dello stesso tipo. Se uno dei parametri di input è `INTEGER`, in ogni caso anche il tipo di restituzione sarà `INTEGER`. I tipi di restituzione validi sono `DECIMAL`, `INT`, `SMALLINT` e `BIGINT`.

Note per l'utilizzo

Puoi utilizzare `%` come operatore di modulo.

Esempi

Per restituire il resto quando un numero viene diviso per un altro, utilizza l'esempio seguente.

```
SELECT MOD(10, 4);

+-----+
| mod |
+-----+
|   2 |
+-----+
```

Per restituire un risultato `DECIMAL` quando utilizzi la funzione MOD, utilizza l'esempio seguente.

```
SELECT MOD(10.5, 4);

+-----+
| mod |
+-----+
| 2.5 |
+-----+
```

Per convertire un numero prima di eseguire la funzione MOD, utilizza l'esempio seguente. Per ulteriori informazioni, consulta Funzione CAST.

```
SELECT MOD(CAST(16.4 AS INTEGER), 5);

+-----+
| mod |
+-----+
|   1 |
+-----+
```

Per controllare se il primo parametro è pari dividendolo per 2, utilizza l'esempio seguente.

```
SELECT mod(5,2) = 0 AS is_even;

+---------+
| is_even |
+---------+
| false   |
```

```
+---------+
```

Per utilizzare % come operatore di modulo, utilizza l'esempio seguente.

```
SELECT 11 % 4 as remainder;

 +-----------+
| remainder |
+-----------+
|         3 |
+-----------+
```

L'esempio seguente utilizza il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire informazioni per le categorie dispari nella tabella CATEGORY, utilizza l'esempio seguente.

```
SELECT catid, catname
FROM category
WHERE MOD(catid,2)=1
ORDER BY 1,2;

+-------+-----------+
| catid |  catname  |
+-------+-----------+
|     1 | MLB       |
|     3 | NFL       |
|     5 | MLS       |
|     7 | Plays     |
|     9 | Pop       |
|    11 | Classical |
+-------+-----------+
```

Funzione PI

La funzione PI restituisce il valore di pi con 14 posizioni decimali.

Sintassi

```
PI()
```

Tipo restituito

DOUBLE PRECISION

Esempi

Per restituire il valore di pi, utilizza l'esempio seguente.

```
SELECT PI();

+-------------------+
|        pi         |
+-------------------+
| 3.141592653589793 |
+-------------------+
```

Funzione POWER

La funzione POWER è una funzione esponenziale che eleva un'espressione numerica alla potenza di una seconda espressione numerica. Ad esempio, 2 alla terza è calcolato come POWER(2,3), con risultato 8.

Sintassi

```
{POW | POWER}(expression1, expression2)
```

Argomenti

expression1

Espressione numerica da elevare. Deve essere un tipo di dati INTEGER, DECIMAL o FLOAT.

expression2

Potenza da elevare expression1. Deve essere un tipo di dati INTEGER, DECIMAL o FLOAT.

Tipo restituito

DOUBLE PRECISION

Esempi

Gli esempi seguenti utilizzano il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Nell'esempio seguente, la funzione POWER viene utilizzata per prevedere quale sarà la vendita dei biglietti nei prossimi 10 anni, in base al numero di biglietti venduti nel 2008 (il risultato della sottoquery). Il tasso di crescita è fissato al 7% all'anno in questo esempio.

```
SELECT (SELECT SUM(qtysold) FROM sales, date
WHERE sales.dateid=date.dateid
AND year=2008) * POW((1+7::FLOAT/100),10) qty2010;

+-------------------+
|      qty2010      |
+-------------------+
| 679353.7540885945 |
+-------------------+
```

L'esempio seguente è una variazione dell'esempio precedente, con un tasso di crescita del 7% all'anno ma con l'intervallo impostato su mesi (120 mesi su 10 anni).

```
SELECT (SELECT SUM(qtysold) FROM sales, date
WHERE sales.dateid=date.dateid
AND year=2008) * POW((1+7::FLOAT/100/12),120) qty2010;

+-----------------+
|     qty2010     |
+-----------------+
| 694034.54678046 |
+-----------------+
```

Funzioni RADIANS

La funzione RADIANS converte un angolo in gradi nel suo equivalente in radianti.

Sintassi

```
RADIANS(number)
```

Argomento

numero

Il parametro di input è un numero DOUBLE PRECISION.

Tipo restituito

DOUBLE PRECISION

Esempi

Per restituire l'equivalente in radianti di 180 gradi, utilizza l'esempio seguente.

```
SELECT RADIANS(180);

+-------------------+
|      radians      |
+-------------------+
| 3.141592653589793 |
+-------------------+
```

Funzione RANDOM

La funzione RANDOM genera un valore casuale compreso tra 0.0 (incluso) e 1.0 (escluso).

Sintassi

```
RANDOM()
```

Tipo restituito

DOUBLE PRECISION

Note per l'utilizzo

Chiamare RANDOM dopo aver impostato un valore di inizializzazione con il comando SET per far sì che RANDOM generi numeri in una sequenza prevedibile.

Esempi

Per calcolare un valore casuale compreso tra 0 e 99, utilizza l'esempio seguente. Se il numero casuale è da 0 a 1, questa query produce un numero casuale compreso tra 0 e 100.

```
SELECT CAST(RANDOM() * 100 AS INT);

+------+
| int4 |
+------+
|   59 |
+------+
```

Questo esempio utilizza il comando SET per impostare il valore SEED in modo tale che RANDOM generi una sequenza prevedibile di numeri.

Per restituire tre numeri interi RANDOM senza impostare il valore SEED, utilizza l'esempio seguente.

```
SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
|    6 |
+------+

SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
|   68 |
+------+

SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
|   56 |
+------+
```

Per impostare il valore SEED su `.25` e restituire altri tre numeri RANDOM, utilizza l'esempio seguente.

```
SET SEED TO .25;
SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
```

```
|   21 |
+------+

SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
|   79 |
+------+

SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
|   12 |
+------+
```

Per ripristinare il valore SEED su `.25` e verificare che RANDOM restituisca gli stessi risultati delle tre chiamate precedenti, utilizza l'esempio seguente.

```
SET SEED TO .25;
SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
|   21 |
+------+

SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
|   79 |
+------+

SELECT CAST(RANDOM() * 100 AS INT);
+------+
| int4 |
+------+
|   12 |
+------+
```

Gli esempi seguenti utilizzano il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per recuperare un campione casuale uniforme di 10 elementi dalla tabella SALES, utilizza l'esempio seguente.

```
SELECT *
FROM sales
ORDER BY RANDOM()
LIMIT 10;

+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
| salesid | listid | sellerid | buyerid | eventid | dateid | qtysold | pricepaid |
 commission |      saletime       |
+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
|   45422 |  51114 |     5983 |   24482 |    4369 |   2118 |       1 |       195 |
 29.25 | 2008-10-19 05:20:07 |
|   42481 |  47638 |     4573 |    6198 |    6479 |   1987 |       4 |      1140 |
   171 | 2008-06-10 09:39:19 |
|   31494 |  34759 |    18895 |    4719 |    7753 |   2090 |       4 |      1024 |
 153.6 | 2008-09-21 03:44:26 |
|  119388 | 136685 |    21815 |   41905 |    2071 |   1884 |       1 |       359 |
 53.85 | 2008-02-27 10:43:10 |
|  166990 | 225037 |    18529 |    7628 |     746 |   2113 |       1 |      2009 |
 301.35 | 2008-10-14 10:07:44 |
|   11146 |  12096 |    42685 |    6619 |    1876 |   2123 |       1 |        29 |
  4.35 | 2008-10-24 06:23:54 |
|  148537 | 172056 |    15102 |   11787 |    6122 |   1923 |       2 |       480 |
    72 | 2008-04-07 03:58:23 |
|   68945 |  78387 |     7359 |   18323 |    6636 |   1910 |       1 |       457 |
 68.55 | 2008-03-25 08:31:03 |
|   52796 |  59576 |     9909 |   15102 |    7958 |   1951 |       1 |       479 |
 71.85 | 2008-05-05 02:25:08 |
|   90684 | 103522 |    38052 |   21549 |    7384 |   2117 |       1 |       313 |
 46.95 | 2008-10-18 05:43:11 |
+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
```

Per recuperare un esempio casuale di 10 elementi, ma sceglierli in proporzione al loro prezzo, utilizza l'esempio seguente. Ad esempio, un elemento il cui prezzo è il doppio di un altro ha il doppio delle probabilità di apparire nei risultati della query.

```
SELECT *
FROM sales
ORDER BY -LOG(RANDOM()) / pricepaid
LIMIT 10;

+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
| salesid | listid | sellerid | buyerid | eventid | dateid | qtysold | pricepaid |
 commission |      saletime       |
+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
|  158340 | 208208 |    17082 |   42018 |    1211 |   2160 |       4 |      6852 |
 1027.8 | 2008-11-30 12:21:43 |
|   53250 |  60069 |    12644 |    7066 |    7942 |   1838 |       4 |      1528 |
 229.2 | 2008-01-12 11:24:56 |
|   22929 |  24938 |    47314 |    6503 |     179 |   2000 |       3 |       741 |
 111.15 | 2008-06-23 08:04:50 |
|  164980 | 221181 |     1949 |   19670 |    1471 |   1906 |       1 |      1330 |
 199.5 | 2008-03-21 07:59:51 |
|  159641 | 211179 |    44897 |   16652 |    7458 |   2128 |       1 |      1019 |
 152.85 | 2008-10-29 02:02:15 |
|   73143 |  83439 |     5716 |    5727 |    7314 |   1903 |       1 |       248 |
  37.2 | 2008-03-18 11:07:42 |
|   84778 |  96749 |    46608 |   32980 |    3883 |   1999 |       2 |       958 |
 143.7 | 2008-06-22 12:13:31 |
|  171096 | 232929 |    43683 |    8536 |    8353 |   1870 |       1 |       929 |
 139.35 | 2008-02-13 01:36:36 |
|   74212 |  84697 |    39809 |   15569 |    5525 |   2105 |       2 |       896 |
 134.4 | 2008-10-06 11:47:50 |
|  158011 | 207556 |    25399 |   16881 |     232 |   2088 |       2 |      2526 |
 378.9 | 2008-09-19 06:00:26 |
+---------+--------+----------+---------+---------+--------+---------+-----------
+------------+---------------------+
```

Funzione ROUND

La funzione ROUND arrotonda i numeri al integer o decimale più vicino.

La funzione ROUND può facoltativamente includere un secondo argomento: `INTEGER` per indicare il numero di cifre decimali per l'arrotondamento, in entrambe le direzioni. Quando non si specifica il secondo argomento, la funzione viene arrotondata al numero intero più vicino. Quando il secondo

argomento specificato è integer, la funzione viene arrotondata al numero più vicino con un valore integer di cifre decimali di precisione.

Sintassi

```
ROUND(number [ , integer ] )
```

Argomenti

numero

Un numero o un'espressione che restituisce un numero. Può essere di tipo DECIMAL, FLOAT8 o SUPER. Amazon Redshift può convertire implicitamente altri tipi di dati numerici.

integer

(Facoltativo) Un valore INTEGER che indica il numero di posizioni decimali per l'arrotondamento, in entrambe le direzioni. Il tipo di dati SUPER non è supportato per questo argomento.

Tipo restituito

ROUND restituisce lo stesso tipo di dati numerici del numero di input.

Quando l'input è di tipo SUPER, l'output mantiene lo stesso tipo dinamico dell'input mentre il tipo statico rimane il tipo SUPER. Quando il tipo dinamico SUPER non è un numero, Amazon Redshift restituisce NULL.

Esempi

Gli esempi seguenti utilizzano il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per arrotondare la commissione pagata per una determinata transazione al numero intero più vicino, utilizza l'esempio seguente.

```
SELECT commission, ROUND(commission)
FROM sales WHERE salesid=10000;

+------------+-------+
| commission | round |
+------------+-------+
```

```
|      28.05 |    28 |
+------------+-------+
```

Per arrotondare la commissione pagata per una determinata transazione alla prima posizione decimale, utilizza l'esempio seguente.

```
SELECT commission, ROUND(commission, 1)
FROM sales WHERE salesid=10000;

+------------+-------+
| commission | round |
+------------+-------+
|      28.05 |  28.1 |
+------------+-------+
```

Per estendere la precisione nella direzione opposta a quella dell'esempio precedente, utilizza l'esempio seguente.

```
SELECT commission, ROUND(commission, -1)
FROM sales WHERE salesid=10000;

+------------+-------+
| commission | round |
+------------+-------+
|      28.05 |    30 |
+------------+-------+
```

Funzione SIN

SIN è una funzione trigonometrica che restituisce il seno di un numero. Il valore restituito è compreso tra -1 e 1.

Sintassi

```
SIN(number)
```

Argomento

numero

Un numero DOUBLE PRECISION in radianti.

Tipo restituito

`DOUBLE PRECISION`

Esempi

Per restituire il seno di `-PI`, utilizza l'esempio seguente.

```
SELECT SIN(-PI());

+-------------------------+
|           sin           |
+-------------------------+
| -0.00000000000000012246 |
+-------------------------+
```

Funzione SIGN

La funzione SIGN restituisce il segno (positivo o negativo) di un numero. Il risultato della funzione SIGN è `1` se l'argomento è positivo,`-1` se l'argomento è negativo, oppure `0` se l'argomento è `0`.

Sintassi

```
SIGN(number)
```

Argomento

numero

Numero o espressione che valuta un numero. Può essere anche di tipo `DECIMAL`, `FLOAT8` o `SUPER`. Altri tipi di dati possono essere convertiti da Amazon Redshift in base alle regole di conversione implicita.

Tipo restituito

SIGN restituisce lo stesso tipo di dati numerici dell'argomento di input. Se l'input è `DECIMAL`, l'output è `DECIMAL(1,0)`.

Quando l'input è di tipo `SUPER`, l'output mantiene lo stesso tipo dinamico dell'input mentre il tipo statico rimane il tipo `SUPER`. Quando il tipo dinamico di `SUPER` non è un numero, Amazon Redshift restituisce un valore `NULL`.

Esempi

L'esempio seguente mostra che la colonna d nella tabella t2 ha il tipo `DOUBLE PRECISION` poiché l'input è `DOUBLE PRECISION` e quella colonna n nella tabella t2 ha `NUMERIC(1,0)` come output poiché l'input è `NUMERIC`.

```
CREATE TABLE t1(d DOUBLE PRECISION, n NUMERIC(12, 2));
INSERT INTO t1 VALUES (4.25, 4.25), (-4.25, -4.25);
CREATE TABLE t2 AS SELECT SIGN(d) AS d, SIGN(n) AS n FROM t1;
SELECT table_name, column_name, data_type FROM SVV_REDSHIFT_COLUMNS WHERE
 table_name='t1' OR table_name='t2';

+------------+-------------+----------------------+
| table_name | column_name |       data_type      |
+------------+-------------+----------------------+
| t1         | d           | double precision     |
| t1         | n           | numeric(12,2)        |
| t2         | d           | double precision     |
| t2         | n           | numeric(1,0)         |
| t1         | col1        | character varying(20) |
+------------+-------------+----------------------+
```

L'esempio seguente utilizza il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per determinare il segno della commissione pagata per una determinata transazione dalla tabella SALES, utilizza l'esempio seguente.

```
SELECT commission, SIGN(commission)
FROM sales WHERE salesid=10000;

+------------+------+
| commission | sign |
+------------+------+
|      28.05 |    1 |
+------------+------+
```

Funzione SQRT

La funzione SQRT restituisce la radice quadrata di un valore `NUMERIC`. La radice quadrata è un numero moltiplicato per sé stesso per ottenere il valore fornito.

Sintassi

```
SQRT(expression)
```

Argomento

espressione

L'espressione deve avere un tipo di dati `INTEGER`, `DECIMAL` o `FLOAT` o un tipo di dati che viene convertito implicitamente in tali tipi di dati. L'espressione può includere funzioni.

Tipo restituito

`DOUBLE PRECISION`

Esempi

Per restituire la radice quadrata di 16, utilizza l'esempio seguente.

```
SELECT SQRT(16);

+------+
| sqrt |
+------+
|    4 |
+------+
```

Per restituire la radice quadrata della stringa 16 utilizzando una conversione di tipo implicito, utilizza l'esempio seguente.

```
SELECT SQRT('16');

+------+
| sqrt |
+------+
|    4 |
+------+
```

Per restituire la radice quadrata di 16,4 dopo aver utilizzato la funzione ROUND, utilizza l'esempio seguente.

```
SELECT SQRT(ROUND(16.4));

+------+
| sqrt |
+------+
|    4 |
+------+
```

Per restituire la lunghezza del raggio quando viene fornita l'area di un cerchio, utilizza l'esempio seguente. Calcola il raggio in pollici, ad esempio, quando viene fornita l'area in pollici quadrati. L'area dell'esempio è 20.

```
SELECT SQRT(20/PI()) AS radius;

+--------------------+
|      radius        |
+--------------------+
| 2.5231325220201604 |
+--------------------+
```

Gli esempi seguenti utilizzano il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire la radice quadrata per i valori COMMISSION dalla tabella SALES, utilizza l'esempio seguente. La colonna COMMISSION è una colonna `DECIMAL`. Questo esempio mostra come utilizzare la funzione in una query con una logica condizionale più complessa.

```
SELECT SQRT(commission)
FROM sales WHERE salesid < 10 ORDER BY salesid;

+--------------------+
|        sqrt        |
+--------------------+
| 10.449880382090505 |
| 3.3763886032268267 |
|  7.245688373094719 |
|  5.123475382979799 |
|  4.806245936279167 |
|  7.687652437513028 |
| 10.871982339941507 |
| 5.4359911699707535 |
```

```
|   9.41541289588513 |
+--------------------+
```

Per restituire la radice quadrata arrotondata per lo stesso set di valori di COMMISSION, utilizza l'esempio seguente.

```
SELECT ROUND(SQRT(commission))
FROM sales WHERE salesid < 10 ORDER BY salesid;

+-------+
| round |
+-------+
|    10 |
|     3 |
|     7 |
|     5 |
|     5 |
|     8 |
|    11 |
|     5 |
|     9 |
+-------+
```

Funzione TAN

TAN è una funzione trigonometrica che restituisce la tangente di un numero. L'argomento di input è un numero (in radianti).

Sintassi

```
TAN(number)
```

Argomento

numero

Un numero `DOUBLE PRECISION`.

Tipo restituito

`DOUBLE PRECISION`

Esempi

Per restituire la tangente di 0, utilizza l'esempio seguente.

```
SELECT TAN(0);

+-----+
| tan |
+-----+
|   0 |
+-----+
```

Funzione TRUNC

La funzione TRUNC tronca i numeri all'intero o al decimale precedente.

La funzione TRUNC può facoltativamente includere un secondo argomento: `INTEGER` per indicare il numero di cifre decimali per l'arrotondamento, in entrambe le direzioni. Quando non si specifica il secondo argomento, la funzione viene arrotondata al numero intero più vicino. Quando il secondo argomento specificato è integer, la funzione viene arrotondata al numero più vicino con un valore integer di cifre decimali di precisione.

Questa funzione può anche troncare un `TIMESTAMP` e restituire `DATE`. Per ulteriori informazioni, consulta Funzione TRUNC.

Sintassi

```
TRUNC(number [ , integer ])
```

Argomenti

numero

Un numero o un'espressione che restituisce un numero. Può essere di tipo `DECIMAL`, `FLOAT8` o `SUPER`. Altri tipi di dati possono essere convertiti da Amazon Redshift in base alle regole di conversione implicita.

integer

(Facoltativo) `INTEGER` che indica il numero di posizioni decimali di precisione, in entrambe le direzioni. Se non viene fornito un valore integer, il numero viene troncato come numero intero; se

viene specificato un valore integer, il numero viene troncato alla posizione decimale specificata. Non è supportato per il tipo di dati SUPER.

Tipo restituito

TRUNC restituisce lo stesso tipo di dati del numero di input.

Quando l'input è di tipo SUPER, l'output mantiene lo stesso tipo dinamico dell'input mentre il tipo statico rimane il tipo SUPER. Quando il tipo dinamico SUPER non è un numero, Amazon Redshift restituisce NULL.

Esempi

La maggior parte degli esempi seguenti usa il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per troncare la commissione pagata per una determinata transazione di vendita, utilizza l'esempio seguente.

```
SELECT commission, TRUNC(commission)
FROM sales WHERE salesid=784;

+------------+-------+
| commission | trunc |
+------------+-------+
|     111.15 |   111 |
+------------+-------+
```

Per troncare lo stesso valore della commissione alla prima posizione decimale, utilizza l'esempio seguente.

```
SELECT commission, TRUNC(commission,1)
FROM sales WHERE salesid=784;

+------------+-------+
| commission | trunc |
+------------+-------+
|     111.15 | 111.1 |
+------------+-------+
```

Per troncare la commissione con un valore negativo per il secondo argomento, utilizza l'esempio seguente. Tieni presente che `111.15` è arrotondato per difetto a `110`.

```
SELECT commission, TRUNC(commission,-1)
FROM sales WHERE salesid=784;

+------------+-------+
| commission | trunc |
+------------+-------+
|     111.15 |   110 |
+------------+-------+
```

Funzioni di oggetti

Di seguito sono riportate le funzioni dell'oggetto SQL supportate da Amazon Redshift per creare oggetti di tipo SUPER:

Argomenti

- Funzione LOWER_ATTRIBUTE_NAMES
- Funzione OBJECT
- Funzione OBJECT_TRANSFORM
- Funzione UPPER_ATTRIBUTE_NAMES

Funzione LOWER_ATTRIBUTE_NAMES

Converte tutti i nomi degli attributi applicabili in un valore SUPER in lettere minuscole, utilizzando la stessa routine di conversione dei maiuscoli di. Funzione LOWER LOWER_ATTRIBUTE_NAMES supporta caratteri UTF-8 multibyte, fino a un massimo di quattro byte per carattere.

Per convertire i nomi degli attributi SUPER in maiuscolo, usa. Funzione UPPER_ATTRIBUTE_NAMES

Sintassi

```
LOWER_ATTRIBUTE_NAMES(super_expression)
```

Argomenti

super_expression

Un'espressione SUPER.

Tipo restituito

SUPER

Note per l'utilizzo

In Amazon Redshift, gli identificatori di colonna tradizionalmente non fanno distinzione tra maiuscole e minuscole e vengono convertiti in lettere minuscole. Se acquisisci dati da formati di dati con distinzione tra maiuscole e minuscole come JSON, i dati potrebbero contenere nomi di attributi composti da maiuscole e minuscole.

Analizza l'esempio seguente.

```
CREATE TABLE t1 (s) AS SELECT JSON_PARSE('{"AttributeName": "Value"}');


SELECT s.AttributeName FROM t1;

attributename
-------------
NULL


SELECT s."AttributeName" FROM t1;

attributename
-------------
NULL
```

Amazon Redshift restituisce NULL per entrambe le query. Per eseguire una query`AttributeName`, usa LOWER_ATTRIBUTE_NAMES per convertire i nomi degli attributi dei dati in lettere minuscole. Analizza l'esempio seguente.

```
CREATE TABLE t2 (s) AS SELECT LOWER_ATTRIBUTE_NAMES(s) FROM t1;
```

```
SELECT s.attributename FROM t2;

attributename
-------------
"Value"


SELECT s.AttributeName FROM t2;

attributename
-------------
"Value"


SELECT s."attributename" FROM t2;

attributename
-------------
"Value"


SELECT s."AttributeName" FROM t2;

attributename
-------------
"Value"
```

Un'opzione correlata per lavorare con nomi di attributi di oggetti con maiuscole e minuscole è l'opzione di `enable_case_sensitive_super_attribute` configurazione, che consente ad Amazon Redshift di riconoscere maiuscole e minuscole nei nomi degli attributi SUPER. Questa può essere una soluzione alternativa all'utilizzo di LOWER_ATTRIBUTE_NAMES. Per ulteriori informazioni su, vai a. `enable_case_sensitive_super_attribute` enable_case_sensitive_super_attribute

Esempi

Conversione dei nomi degli attributi SUPER in lettere minuscole

L'esempio seguente utilizza LOWER_ATTRIBUTE_NAMES per convertire i nomi degli attributi di tutti i valori SUPER in una tabella.

```
-- Create a table and insert several SUPER values.
```

```
CREATE TABLE t (i INT, s SUPER);

INSERT INTO t VALUES
  (1, NULL),
  (2, 'A'::SUPER),
  (3, JSON_PARSE('{"AttributeName": "B"}')),
  (4, JSON_PARSE(
     '[{"Subobject": {"C": "C"},
       "Subarray": [{"D": "D"}, "E"]
      }]'));

-- Convert all attribute names to lowercase.
UPDATE t SET s = LOWER_ATTRIBUTE_NAMES(s);

SELECT i, s FROM t ORDER BY i;

 i |                         s
---+-----------------------------------------------
 1 | NULL
 2 | "A"
 3 | {"attributename":"B"}
 4 | [{"subobject":{"c":"C"},"subarray":[{"d":"D"}, "E"]}]
```

Osservate come funziona LOWER_ATTRIBUTE_NAMES.

- I valori NULL e i valori SCALARI SUPER, come ad esempio, sono invariati. `"A"`
- In un oggetto SUPER, tutti i nomi degli attributi vengono modificati in lettere minuscole, ma i valori degli attributi, ad esempio, rimangono invariati. `"B"`
- LOWER_ATTRIBUTE_NAMES si applica in modo ricorsivo a qualsiasi oggetto SUPER annidato all'interno di un array SUPER o all'interno di un altro oggetto.

Utilizzo di LOWER_ATTRIBUTE_NAMES su un oggetto SUPER con nomi di attributi duplicati

Se un oggetto SUPER contiene attributi i cui nomi differiscono solo in maiuscole e minuscole, LOWER_ATTRIBUTE_NAMES genererà un errore. Analizza l'esempio seguente.

```
SELECT LOWER_ATTRIBUTE_NAMES(JSON_PARSE('{"A": "A", "a": "a"}'));

error:   Invalid input
code:    8001
context: SUPER value has duplicate attributes after case conversion.
```

Funzione OBJECT

Crea un oggetto del tipo di dati SUPER.

Sintassi

```
OBJECT ( [ key1, value1 ], [ key2, value2 ...] )
```

Argomenti

key1, key2

Espressioni che valutano stringhe di tipo VARCHAR.

value1, value2

Espressioni di qualsiasi tipo di dati Amazon Redshift eccetto i tipi datetime, poiché Amazon Redshift non esegue il casting dei tipi datetime nel tipo di dati SUPER. Per ulteriori informazioni sui tipi datetime, consulta Tipi datetime.

Non è necessario che le espressioni `value` in un oggetto siano dello stesso tipo di dati.

Tipo di restituzione

`SUPER`

Esempio

```
-- Creates an empty object.
select object();

object
--------
{}
(1 row)

-- Creates objects with different keys and values.
select object('a', 1, 'b', true, 'c', 3.14);

object
---------------------------
```

```
{"a":1,"b":true,"c":3.14}
(1 row)

select object('a', object('aa', 1), 'b', array(2,3), 'c', json_parse('{}'));

object
---------------------------------
{"a":{"aa":1},"b":[2,3],"c":{}}
(1 row)

-- Creates objects using columns from a table.
create table bar (k varchar, v super);
insert into bar values ('k1', json_parse('[1]')), ('k2', json_parse('{}'));
select object(k, v) from bar;

object
------------
{"k1":[1]}
{"k2":{}}
(2 rows)

-- Errors out because DATE type values can't be converted to SUPER type.
select object('k', '2008-12-31'::date);

ERROR:  OBJECT could not convert type date to super
```

Funzione OBJECT_TRANSFORM

Trasforma un oggetto SUPER.

Sintassi

```
OBJECT_TRANSFORM(
  input
  [KEEP path1, ...]
  [SET
    path1, value1,
    ...,  ...
  ]
)
```

Argomenti

input

Un'espressione che restituisce un oggetto di tipo SUPER.

KEEP

Tutti i valori path specificati in questa clausola vengono mantenuti e passati all'oggetto di output.

Questa clausola è facoltativa.

path1, path2, ...

Valori letterali di stringa costanti, nel formato di componenti del percorso tra virgolette doppie delimitate da punti. Ad esempio, `'"a"."b"."c"'` è un valore di percorso valido. Questo vale per il parametro path in entrambe le clausole KEEP e SET.

SET

Coppie di path e value per modificare un percorso esistente o aggiungerne uno nuovo e impostare il valore del percorso nell'oggetto di output.

Questa clausola è facoltativa.

value1, value2, ...

Espressioni che restituiscono valori di tipo SUPER. Tieni presente che i tipi numerico, testo e booleano restituiscono valori SUPER.

Tipo restituito

SUPER

Note per l'utilizzo

OBJECT_TRANSFORM restituisce un oggetto di tipo SUPER contenente i valori del percorso dell'input specificati in KEEP e le coppie di path e value specificate in SET.

Se sia KEEP che SET sono vuoti, OBJECT_TRANSFORM restituisce l'input.

Se l'input non è un oggetto di tipo SUPER, OBJECT_TRANSFORM restituisce l'input, indipendentemente dai valori KEEP o SET.

Esempio

L'esempio seguente trasforma un oggetto SUPER in un altro oggetto SUPER.

```
CREATE TABLE employees (
    col_person SUPER
);

INSERT INTO employees
VALUES
    (
        json_parse('
            {
                "name": {
                    "first": "John",
                    "last": "Doe"
                },
                "age": 25,
                "ssn": "111-22-3333",
                "company": "Company Inc.",
                "country": "U.S."
            }
        ')
    ),
    (
        json_parse('
            {
                "name": {
                    "first": "Jane",
                    "last": "Appleseed"
                },
                "age": 34,
                "ssn": "444-55-7777",
                "company": "Organization Org.",
                "country": "Ukraine"
            }
        ')
    )
;

SELECT
    OBJECT_TRANSFORM(
        col_person
        KEEP
```

```
            '"name"."first"',
            '"age"',
            '"company"',
            '"country"'
        SET
            '"name"."first"', UPPER(col_person.name.first::TEXT),
            '"age"', col_person.age + 5,
            '"company"', 'Amazon'
    ) AS col_person_transformed
FROM employees;

--This result is formatted for ease of reading.
                  col_person_transformed
-------------------------------------------------------------
{
    "name": {
        "first": "JOHN"
    },
    "age": 30,
    "company": "Amazon",
    "country": "U.S."
}
{
    "name": {
        "first": "JANE"
    },
    "age": 39,
    "company": "Amazon",
    "country": "Ukraine"
}
```

Funzione UPPER_ATTRIBUTE_NAMES

Converte in maiuscolo tutti i nomi degli attributi applicabili in un valore SUPER, utilizzando la stessa routine di conversione dei maiuscoli di. Funzione UPPER UPPER_ATTRIBUTE_NAMES supporta caratteri UTF-8 multibyte, fino a un massimo di quattro byte per carattere.

Per convertire i nomi degli attributi SUPER in lettere minuscole, usa. Funzione LOWER_ATTRIBUTE_NAMES

Sintassi

```
UPPER_ATTRIBUTE_NAMES(super_expression)
```

Argomenti

super_expression

Un'espressione SUPER.

Tipo restituito

SUPER

Esempi

Conversione dei nomi degli attributi SUPER in maiuscolo

L'esempio seguente utilizza UPPER_ATTRIBUTE_NAMES per convertire i nomi degli attributi di tutti i valori SUPER in una tabella.

```
-- Create a table and insert several SUPER values.
CREATE TABLE t (i INT, s SUPER);

INSERT INTO t VALUES
  (1, NULL),
  (2, 'a'::SUPER),
  (3, JSON_PARSE('{"AttributeName": "b"}')),
  (4, JSON_PARSE(
     '[{"Subobject": {"c": "c"},
       "Subarray": [{"d": "d"}, "e"]
      }]'));

-- Convert all attribute names to uppercase.
UPDATE t SET s = UPPER_ATTRIBUTE_NAMES(s);

SELECT i, s FROM t ORDER BY i;

 i |                         s
---+---------------------------------------------------
 1 | NULL
 2 | "a"
 3 | {"ATTRIBUTENAME":"B"}
 4 | [{"SUBOBJECT":{"C":"c"},"SUBARRAY":[{"D":"d"}, "e"]}]
```

Osservate come funziona UPPER_ATTRIBUTE_NAMES.

- I valori NULL e i valori SCALARI SUPER, come ad esempio, sono invariati. `"a"`
- In un oggetto SUPER, tutti i nomi degli attributi vengono modificati in lettere maiuscole, ma i valori degli attributi, ad esempio, rimangono invariati. `"b"`
- UPPER_ATTRIBUTE_NAMES si applica in modo ricorsivo a qualsiasi oggetto SUPER annidato all'interno di un array SUPER o all'interno di un altro oggetto.

Utilizzo di UPPER_ATTRIBUTE_NAMES su un oggetto SUPER con nomi di attributi duplicati

Se un oggetto SUPER contiene attributi i cui nomi differiscono solo in maiuscole e minuscole, UPPER_ATTRIBUTE_NAMES genererà un errore. Analizza l'esempio seguente.

```
SELECT UPPER_ATTRIBUTE_NAMES(JSON_PARSE('{"A": "A", "a": "a"}'));

error:   Invalid input
code:    8001
context: SUPER value has duplicate attributes after case conversion.
```

Funzioni spaziali

Le relazioni tra oggetti geometrici si basano sul modello Dimensionally Extended nine-Intersection Model (DE-9IM). Questo modello definisce predicati come uguale, contiene e ricopre. Per ulteriori informazione sulla definizione delle relazioni spaziali, consultare DE-9IM su Wikipedia.

Per ulteriori informazioni su come utilizzare i dati spaziali con Amazon Redshift, consulta Query su dati spaziali in Amazon Redshift.

Amazon Redshift fornisce funzioni spaziali che funzionano con i tipi di dati `GEOMETRY` e `GEOGRAPHY`. Di seguito sono elencate le funzioni che supportano il tipo di dati `GEOGRAPHY`:

- ST_Area
- ST_AsEWKT
- ST_ JSON AsGeo
- ST_ EWKB AsHex
- ST_ WKB AsHex
- ST_ AsText
- ST_Distance

- ST_ GeogFromText
- GeogFromST_ WKB
- ST_Length
- ST_NPoints
- ST_Perimeter

Di seguito viene elencato il set completo di funzioni spaziali supportate da Amazon Redshift.

Argomenti

- AddBBox
- DropBBox
- GeometryType
- H3_ FromLongLat
- H3_ FromPoint
- H3_Polyfill
- ST_ AddPoint
- ST_Angle
- ST_Area
- ST_ AsBinary
- ST_AsEWKB
- ST_AsEWKT
- ST_ JSON AsGeo
- ST_ WKB AsHex
- ST_ EWKB AsHex
- ST_ AsText
- ST_Azimuth
- ST_Boundary
- ST_Buffer
- ST_Centroid
- ST_Collect
- ST_Contains

- ST_ ContainsProperly
- ST_ ConvexHull
- ST_ CoveredBy
- ST_Covers
- ST_Crosses
- ST_Dimension
- ST_Disjoint
- ST_Distance
- ST_ DistanceSphere
- ST_DWithin
- ST_ EndPoint
- ST_Envelope
- ST_Equals
- ST_ ExteriorRing
- ST_Force2D
- ST_Force3D
- ST_Force3DM
- ST_Force3DZ
- ST_Force4D
- ST_ GeoHash
- ST_ GeogFromText
- GeogFromST_ WKB
- ST_GeometryN
- ST_ GeometryType
- ST_ EWKB GeomFrom
- ST_ GeomFrom EWKT
- ST_ GeomFromGeoHash
- GeomFromGeoST_ JSON
- ST_ GeomFromGeoSquare
- ST_ GeomFromText

- GeomFromST_ WKB
- ST_ GeoSquare
- ST_ N InteriorRing
- ST_Intersects
- ST_Intersection
- IsPolygonST_ CCW
- IsPolygonST_ CW
- ST_ IsClosed
- ST_ IsCollection
- ST_ IsEmpty
- ST_ IsRing
- ST_ IsSimple
- ST_ IsValid
- ST_Length
- ST_ LengthSphere
- ST_Length2D
- ST_ LineFromMultiPoint
- ST_ LineInterpolatePoint
- ST_M
- ST_ MakeEnvelope
- ST_ MakeLine
- ST_ MakePoint
- ST_ MakePolygon
- ST_ MemSize
- ST_MMax
- ST_MMin
- ST_Multi
- ST_NDims
- ST_NPoints
- ST_NRings

- ST_ NumGeometries
- ST_ NumInteriorRings
- ST_ NumPoints
- ST_Perimeter
- ST_Perimeter2D
- ST_Point
- ST_PointN
- ST_Points
- ST_Polygon
- ST_ RemovePoint
- ST_Reverse
- ST_ SetPoint
- ST_SetSRID
- ST_Simplify
- ST_SRID
- ST_ StartPoint
- ST_Touches
- ST_Transform
- ST_Union
- ST_Within
- ST_X
- ST_XMax
- ST_XMin
- ST_Y
- ST_YMax
- ST_YMin
- ST_Z
- ST_ZMax
- ST_ZMin
- SupportsBBox

AddBBox

AddBBox restituisce una copia della geometria di input che supporta la codifica con un bounding box precalcolato. Per ulteriori informazioni sul supporto per i bounding box, consultare Riquadro di delimitazione.

Sintassi

```
AddBBox(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

GEOMETRY

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente codice SQL restituisce una copia di una geometria poligonale di input che supporta la codifica con un bounding box.

```
SELECT ST_AsText(AddBBox(ST_GeomFromText('POLYGON((0 0,1 0,0 1,0 0))')));
```

```
 st_astext
----------
 POLYGON((0 0,1 0,0 1,0 0))
```

DropBBox

DropBBox restituisce una copia della geometria di input che supporta la codifica con un bounding box precalcolato. Per ulteriori informazioni sul supporto per i bounding box, consultare Riquadro di delimitazione.

Sintassi

```
DropBBox(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

GEOMETRY

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente codice SQL restituisce una copia di una geometria poligonale di input che non supporta la codifica con un bounding box.

```
SELECT ST_AsText(DropBBox(ST_GeomFromText('POLYGON((0 0,1 0,0 1,0 0))')));
```

```
 st_astext
----------
 POLYGON((0 0,1 0,0 1,0 0))
```

GeometryType

GeometryType restituisce il sottotipo di una geometria di input come stringa.

Sintassi

```
GeometryType(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

VARCHAR che rappresenta il sottotipo di geom.

Se geom è nullo, allora viene restituito il valore nullo.

I valori restituiti sono i seguenti.

Valore stringa restituito per geometrie 2D, 3DZ, 4D	Valore stringa restituito per geometrie 3DM	Sottotipo dato di tipo geometry
POINT	POINTM	Restituito se geom è di sottotipo POINT
LINESTRING	LINESTRINGM	Restituito se geom è di sottotipo LINESTRING
POLYGON	POLYGONM	Restituito se geom è di sottotipo POLYGON
MULTIPOINT	MULTIPOINTM	Restituito se geom è di sottotipo MULTIPOINT
MULTILINESTRING	MULTILINESTRINGM	Restituito se geom è di sottotipo MULTILINESTRING
MULTIPOLYGON	MULTIPOLYGONM	Restituito se geom è di sottotipo MULTIPOLYGON
GEOMETRYCOLLECTION	GEOMETRYCOLLECTIONM	Restituito se geom è di sottotipo GEOMETRYCOLLECTION

Esempi

Il seguente comando SQL converte una rappresentazione in formato Well-Known Text (WKT) di un poligono e restituisce il sottotipo GEOMETRY come stringa.

```
SELECT GeometryType(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'));
```

```
geometrytype
-------------
 POLYGON
```

H3_ FromLongLat

H3_ FromLongLat restituisce l'ID della cella H3 corrispondente da longitudine, latitudine e risoluzione di input. Per ulteriori informazioni sull'indicizzazione H3, consulta H3.

Sintassi

```
H3_FromLongLat(longitude, lattitude, resolution)
```

Argomenti

longitudine

Un valore di tipo DOUBLE PRECISION o un'espressione che restituisce un valore di tipo DOUBLE PRECISION.

latitudine

Un valore di tipo DOUBLE PRECISION o un'espressione che restituisce un valore di tipo DOUBLE PRECISION.

risoluzione

Un valore di tipo INTEGER o un'espressione che restituisce un valore di tipo INTEGER. Il valore rappresenta la risoluzione del sistema a griglia H3. Il valore deve essere un numero intero compreso tra 0 e 15, di cui 0 è la risoluzione più grossolana e 15 la più granulare.

Tipo restituito

BIGINT: rappresenta l'ID della cella H3.

Se la risoluzione non è compresa nell'intervallo valido viene restituito un errore.

Esempi

La seguente istruzione SQL restituisce l'ID della cella H3 in base a longitudine 0, latitudine 0 e risoluzione 10.

```
SELECT H3_FromLongLat(0, 0, 10);
```

```
 h3_fromlonglat
-------------------
 623560421467684863
```

H3_ FromPoint

H3_ FromPoint restituisce l'ID della cella H3 corrispondente da un punto di geometria e una risoluzione di input. Per ulteriori informazioni sull'indicizzazione H3, consulta H3.

Sintassi

```
H3_FromPoint(geom, resolution)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. L'argomento geom deve essere POINT.

risoluzione

Un valore di tipo INTEGER o un'espressione che restituisce un valore di tipo INTEGER. Il valore rappresenta la risoluzione del sistema a griglia H3. Il valore deve essere un numero intero compreso tra 0 e 15, di cui 0 è la risoluzione più grossolana e 15 la più granulare.

Tipo restituito

BIGINT: rappresenta l'ID della cella H3.

Se geom non è un POINT, allora viene restituito un errore.

Se la risoluzione non è compresa nell'intervallo valido viene restituito un errore.

Se geom è vuoto, viene restituito NULL.

Esempi

La seguente istruzione SQL restituisce l'ID della cella H3 dal punto 0,0 e risoluzione 10.

```
SELECT H3_FromPoint(ST_GeomFromText('POINT(0 0)'), 10);
```

```
 h3_frompoint
------------------
 623560421467684863
```

H3_Polyfill

H3_Polyfill restituisce l'ID della cella H3 corrispondente agli esagoni e ai pentagoni contenuti nel poligono di input della risoluzione data. Per ulteriori informazioni sull'indicizzazione H3, consulta H3.

Sintassi

```
H3_Polyfill(geom, resolution)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. L'argomento geom deve essere POLYGON.

risoluzione

Un valore di tipo INTEGER o un'espressione che restituisce un valore di tipo INTEGER. Il valore rappresenta la risoluzione del sistema a griglia H3. Il valore deve essere un numero intero compreso tra 0 e 15, di cui 0 è la risoluzione più grossolana e 15 la più granulare.

Tipo restituito

SUPER: rappresenta un elenco di ID di celle H3.

Se geom non è un POLYGON, allora viene restituito un errore.

Se la risoluzione non è compresa nell'intervallo valido viene restituito un errore.

Se geom è vuoto, viene restituito NULL.

Esempi

La seguente istruzione SQL restituisce un array di ID di celle H3 con tipo di dati SUPER da un poligono e una risoluzione 4.

```
SELECT H3_Polyfill(ST_GeomFromText('POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))'), 4);
```

```
 h3_polyfill
-------------------------------------------------------------------------------------------------
 [596538848238895103,596538805289222143,596538856828829695,596538813879156735,5965379205259591
```

ST_ AddPoint

ST_ AddPoint restituisce una geometria di stringhe di linee che è la stessa della geometria di input con un punto aggiunto. Se viene fornito un indice, il punto viene aggiunto alla posizione dell'indice. Se l'indice è -1 o non viene fornito, il punto viene aggiunto alla linestring.

L'indice è a base zero. L'identificatore del sistema di riferimento spaziale (SRID) del risultato è lo stesso della geometria di input.

La dimensione della geometria restituita è la stessa del valore geom1. Se geom1 e geom2 hanno dimensioni diverse, geom2 è proiettato sulla dimensione di geom1.

Sintassi

```
ST_AddPoint(geom1, geom2)
```

```
ST_AddPoint(geom1, geom2, index)
```

Argomenti

geom1

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo sottotipo deve essere `LINESTRING`.

geom2

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo sottotipo deve essere `POINT`. Il punto può essere il punto vuoto.

indice

Valore del tipo di dati `INTEGER` che rappresenta la posizione di un indice a base zero.

Tipo restituito

`GEOMETRY`

Se geom1, geom2 o index sono nulli, allora viene restituito il valore nullo.

Se geom2 è il punto vuoto, quindi viene restituita una copia di geom1.

Se geom1 non è un `LINESTRING`, allora viene restituito un errore.

Se geom2 non è un `POINT`, allora viene restituito un errore.

Se index è fuori intervallo, viene restituito un errore. I valori validi per la posizione dell'indice sono -1 o un valore compreso tra 0 e `ST_NumPoints(geom1)`.

Esempi

Il seguente codice SQL aggiunge un punto a una linestring per renderla una linestring chiusa.

```
WITH tmp(g) AS (SELECT ST_GeomFromText('LINESTRING(0 0,10 0,10 10,5 5,0 5)',4326))
SELECT ST_AsEWKT(ST_AddPoint(g, ST_StartPoint(g))) FROM tmp;
```

```
 st_asewkt
-------------------------------------------------
 SRID=4326;LINESTRING(0 0,10 0,10 10,5 5,0 5,0 0)
```

Il seguente codice SQL aggiunge un punto a una posizione specifica in una linestring.

```
WITH tmp(g) AS (SELECT ST_GeomFromText('LINESTRING(0 0,10 0,10 10,5 5,0 5)',4326))
SELECT ST_AsEWKT(ST_AddPoint(g, ST_SetSRID(ST_Point(5, 10), 4326), 3)) FROM tmp;
```

```
 st_asewkt
-------------------------------------------------
 SRID=4326;LINESTRING(0 0,10 0,10 10,5 10,5 5,0 5)
```

ST_Angle

ST_Angle restituisce l'angolo in radianti tra i punti misurati in senso orario.

- Se vengono inseriti tre punti, l'angolo restituito P1-P2-P3 viene misurato come se fosse ottenuto ruotando da P1 a P3 attorno a P2 in senso orario.
- Se vengono inseriti quattro punti, viene restituito l'angolo restituito in senso orario formato dalle linee dirette P1-P2 e P3-P4. Se l'input è un caso degenerato (ovvero, P1 è uguale a P2 o P3 uguale a P4), viene restituito null.

Il valore di restituzione è in radianti ed è compreso nell'intervallo [0, 2π).

ST_Angle opera sulle proiezioni 2D delle geometrie di input.

Sintassi

```
ST_Angle(geom1, geom2, geom3)
```

```
ST_Angle(geom1, geom2, geom3, geom4)
```

Argomenti

geom1

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo sottotipo deve essere `POINT`.

geom2

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo sottotipo deve essere `POINT`.

geom3

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo sottotipo deve essere `POINT`.

geom4

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo sottotipo deve essere `POINT`.

Tipo restituito

`DOUBLE PRECISION`.

Se geom1 è uguale a geom2 o geom2 è uguale a geom3, allora viene restituito un valore null.

Se geom1, geom2, geom3 o geom4 è null, allora viene restituito un valore null.

Se uno di geom1, geom2, geom3 o geom4 è il punto vuoto, allora viene restituito un errore.

Se geom1, geom2, geom3 e geom4 non hanno lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito un errore.

Esempi

Il seguente SQL restituisce l'angolo convertito in gradi di tre punti di input.

```
SELECT ST_Angle(ST_Point(1,1), ST_Point(0,0), ST_Point(1,0)) / Pi() * 180.0 AS angle;
```

```
 angle
---------------
    45
```

Il seguente SQL restituisce l'angolo convertito in gradi di quattro punti di input.

```
SELECT ST_Angle(ST_Point(1,1), ST_Point(0,0), ST_Point(1,0), ST_Point(2,0)) / Pi() *
 180.0 AS angle;
```

```
 angle
---------------
   225
```

ST_Area

Per una geometria di input, ST_Area restituisce l'area cartesiana della proiezione 2D. Le unità di area sono le stesse delle unità in cui sono espresse le coordinate della geometria di input. Per punti, linee di linea, multipunti e multilinestring, la funzione restituisce 0. Per le raccolte di geometria, restituisce la somma delle aree delle geometrie nella raccolta.

Per una geografia di input, ST_Area restituisce l'area geodetica della proiezione 2D di una geografia areale di ingresso calcolata sullo sferoide determinato dallo SRID. L'unità di lunghezza è in metri quadrati. La funzione restituisce zero (0) per punti, multipunti e geografie lineari. Quando l'input è una raccolta di geometrie, la funzione restituisce la somma delle aree delle geografie areali nella raccolta.

Sintassi

```
ST_Area(geo)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

Tipo restituito

DOUBLE PRECISION

Se geo è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente SQL restituisce l'area cartesiana di un multipoligono.

```
SELECT ST_Area(ST_GeomFromText('MULTIPOLYGON(((0 0,10 0,0 10,0 0)),((10 0,20 0,20 10,10
 0)))'));
```

```
 st_area
---------
     100
```

Il seguente SQL restituisce l'area di un poligono in una geografia.

```
SELECT ST_Area(ST_GeogFromText('polygon((34 35, 28 30, 25 34, 34 35))'));
```

```
     st_area
------------------
```

```
 201824655743.383
```

Il seguente SQL restituisce zero per una geografia lineare.

```
SELECT ST_Area(ST_GeogFromText('multipoint(0 0, 1 1, -21.32 121.2)'));
```

```
 st_area
---------
       0
```

ST_ AsBinary

ST_ AsBinary restituisce la rappresentazione esadecimale nota binaria (WKB) di una geometria di input. Per le geometrie 3DZ, 3DM e 4D, ST_ AsBinary utilizza il valore standard Open Geospatial Consortium (OGC) per il tipo di geometria.

Sintassi

```
ST_AsBinary(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`VARBYTE`

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce la rappresentazione esadecimale WKB di un poligono.

```
SELECT ST_AsBinary(ST_GeomFromText('POLYGON((0 0,0 1,1 1,1 0,0 0))',4326));
```

```
st_asbinary
```

```
--------------------------------
0103000000010000000500000000000000000000000000000000000000000000000000000000000000000000000000000000F03F00000
```

ST_AsEWKB

ST_AsEWKB restituisce la rappresentazione Extended Well-Known Binary (EWKB) di una geometria di input. Per le geometrie 3DZ, 3DM e 4D, ST_AsEWKB utilizza il valore standard OGC (Open Geospatial Consortium) per il tipo di geometria.

Sintassi

```
ST_AsEWKB(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

VARBYTE

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce la rappresentazione esadecimale EWKB di un poligono.

```
SELECT ST_AsEWKB(ST_GeomFromText('POLYGON((0 0,0 1,1 1,1 0,0 0))',4326));
```

```
st_asewkb
--------------------------------
0103000020E6100000010000000500000000000000000000000000000000000000000000000000000000000000000000000000000000F
```

ST_AsEWKT

ST_AsEWKT restituisce la rappresentazione Extended Well-Known Text (EWKT) di una geometria o geografia di input. Per le geometrie 3DZ, 3DM e 4D, ST_AsEWKB aggiunge Z, M o ZM al valore WKT per il tipo di geometria.

Sintassi

```
ST_AsEWKT(geo)
```

```
ST_AsEWKT(geo, precision)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

precisione

Un valore di tipo INTEGER. Per le geometrie, le coordinate di geo vengono visualizzate utilizzando la precisione specificata 1-20. Se la precisione non è specificata, viene usato il valore predefinito 15. Per le geografie, le coordinate di geo vengono visualizzate utilizzando la precisione specificata. Se la precisione non è specificata, viene usato il valore predefinito 15.

Tipo restituito

VARCHAR

Se geo è nullo, allora viene restituito il valore nullo.

Se precision è nullo, allora viene restituito il valore nullo.

Se il risultato è una VARCHAR di dimensioni maggiori di 64 KB, viene restituito un errore.

Esempi

Il seguente comando SQL restituisce la rappresentazione in formato EWKT di una linestring.

```
SELECT ST_AsEWKT(ST_GeomFromText('LINESTRING(3.141592653589793
 -6.283185307179586,2.718281828459045 -1.414213562373095)', 4326));
```

```
st_asewkt
--------------------------------
```

```
 SRID=4326;LINESTRING(3.14159265358979 -6.28318530717959,2.71828182845905
 -1.41421356237309)
```

Il seguente comando SQL restituisce la rappresentazione in formato EWKT di una linestring. Le coordinate delle geometrie sono visualizzate con una precisione di sei cifre.

```
SELECT ST_AsEWKT(ST_GeomFromText('LINESTRING(3.141592653589793
 -6.283185307179586,2.718281828459045 -1.414213562373095)', 4326), 6);
```

```
st_asewkt
------------------------------
 SRID=4326;LINESTRING(3.14159 -6.28319,2.71828 -1.41421)
```

Il seguente SQL restituisce la rappresentazione EWKT di una geografia.

```
SELECT ST_AsEWKT(ST_GeogFromText('LINESTRING(110 40, 2 3, -10 80, -7 9)'));
```

```
                 st_asewkt
-------------------------------------------
 SRID=4326;LINESTRING(110 40,2 3,-10 80,-7 9)
```

ST_ JSON AsGeo

ST_ AsGeo JSON restituisce la rappresentazione GeoJSON di una geometria o geografia di input. Per ulteriori informazioni su GeoJSON, consultare la voce GeoJSON su Wikipedia.

Per le geometrie 3DZ e 4D, la geometria di uscita è una proiezione 3DZ della geometria 3DZ o 4D di input. In altre parole, le coordinate x, y e z sono presenti nell'output. Per le geometrie 3DM, la geometria di output è una proiezione 2D della geometria 3DM di input. In altre parole, le coordinate x e y sono presenti nell'output.

Per le aree geografiche di input, ST_ AsGeo JSON restituisce la rappresentazione GeoJSON di una geografia di input. Le coordinate della geografia vengono visualizzate utilizzando la precisione specificata.

Sintassi

```
ST_AsGeoJSON(geo)
```

```
ST_AsGeoJSON(geo, precision)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

precisione

Un valore di tipo INTEGER. Per le geometrie, le coordinate di geo vengono visualizzate utilizzando la precisione specificata 1-20. Se la precisione non è specificata, viene usato il valore predefinito 15. Per le geografie, le coordinate di geo vengono visualizzate utilizzando la precisione specificata. Se la precisione non è specificata, viene usato il valore predefinito 15.

Tipo restituito

VARCHAR

Se geo è nullo, allora viene restituito il valore nullo.

Se precision è nullo, allora viene restituito il valore nullo.

Se il risultato è una VARCHAR di dimensioni maggiori di 64 KB, viene restituito un errore.

Esempi

Il seguente comando SQL restituisce la rappresentazione GeoJSON di una linestring.

```
SELECT ST_AsGeoJSON(ST_GeomFromText('LINESTRING(3.141592653589793
 -6.283185307179586,2.718281828459045 -1.414213562373095)'));
```

```
st_asgeojson
-------------------------------------------------------------------------------------------
 {"type":"LineString","coordinates":[[3.14159265358979,-6.28318530717959],
[2.71828182845905,-1.4142135623709]]}
```

Il seguente comando SQL restituisce la rappresentazione GeoJSON di una linestring. Le coordinate delle geometrie sono visualizzate con una precisione di sei cifre.

```
SELECT ST_AsGeoJSON(ST_GeomFromText('LINESTRING(3.141592653589793
 -6.283185307179586,2.718281828459045 -1.414213562373095)'), 6);
```

```
st_asgeojson
--------------------------------------------------------------------------------
  {"type":"LineString","coordinates":[[3.14159,-6.28319],[2.71828,-1.41421]]}
```

Il seguente comando SQL restituisce la rappresentazione GeoJSON di una geografia.

```
SELECT ST_AsGeoJSON(ST_GeogFromText('LINESTRING(110 40, 2 3, -10 80, -7 9)'));
```

```
                             st_asgeojson
--------------------------------------------------------------------------
 {"type":"LineString","coordinates":[[110,40],[2,3],[-10,80],[-7,9]]}
```

ST_ WKB AsHex

ST_ AsHex WKB restituisce la rappresentazione binaria esadecimale nota (WKB) di una geometria o geografia di input utilizzando caratteri esadecimali ASCII (0—9, A—F). Per le geometrie o le aree geografiche 3DZ, 3DM e 4D, ST_ WKB utilizza il valore standard Open Geospatial Consortium (OGC) per il tipo di geometria o geografia. AsHex

Sintassi

```
ST_AsHexWKB(geo)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

Tipo restituito

VARCHAR

Se geo è nullo, allora viene restituito il valore nullo.

Se il risultato è una VARCHAR di dimensioni maggiori di 64 KB, viene restituito un errore.

Esempi

Il seguente comando SQL restituisce la rappresentazione esadecimale WKB di un poligono in una geometria.

```
SELECT ST_AsHexWKB(ST_GeomFromText('POLYGON((0 0,0 1,1 1,1 0,0 0))',4326));
```

```
st_ashexwkb
--------------------------------
0103000000010000000500000000000000000000000000000000000000000000000000F03F00000
```

Il seguente comando SQL restituisce la rappresentazione esadecimale WKB di un poligono in una geografia.

```
SELECT ST_AsHexWKB(ST_GeogFromText('POLYGON((0 0,0 1,1 1,1 0,0 0))'));
```

```
st_ashexwkb
--------------------------------
0103000000010000000500000000000000000000000000000000000000000000000000F03F00000
```

ST_ EWKB AsHex

ST_ AsHex EWKB restituisce la rappresentazione binaria estesa ben nota (EWKB) di una geometria o di una geografia di input utilizzando caratteri esadecimali ASCII (0-9, A—F). Per le geometrie o le aree geografiche 3DZ, 3DM e 4D, ST_ AsHex EWKB utilizza il valore WKB esteso di PostGIS per la geometria o il tipo di geografia.

Sintassi

```
ST_AsHexEWKB(geo)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

Tipo restituito

VARCHAR

Se geo è nullo, allora viene restituito il valore nullo.

Se il risultato è una VARCHAR di dimensioni maggiori di 64 KB, viene restituito un errore.

Esempi

Il seguente comando SQL restituisce la rappresentazione esadecimale EWKB di un poligono in una geometria.

```
SELECT ST_AsHexEWKB(ST_GeomFromText('POLYGON((0 0,0 1,1 1,1 0,0 0))',4326));
```

```
st_ashexewkb
--------------------------------
0103000020E6100000010000000500000000000000000000000000000000000000000000000000F0
```

Il seguente comando SQL restituisce la rappresentazione esadecimale EWKB di un poligono in una geografia.

```
SELECT ST_AsHexEWKB(ST_GeogFromText('POLYGON((0 0,0 1,1 1,1 0,0 0))'));
```

```
st_ashexewkb
--------------------------------
0103000020E6100000010000000500000000000000000000000000000000000000000000000000F0
```

ST_ AsText

ST_ AsText restituisce la rappresentazione di testo noto (WKT) di una geometria o geografia di input. Per le geometrie o geografie 3DZ, 3DM e 4D, ST_AsEWKB aggiunge Z, M o ZM al valore WKT per il tipo di geometria o geografia.

Sintassi

```
ST_AsText(geo)
```

```
ST_AsText(geo, precision)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

precisione

Un valore di tipo INTEGER. Per le geometrie, le coordinate di geo vengono visualizzate utilizzando la precisione specificata 1-20. Se la precisione non è specificata, viene usato il valore predefinito 15. Per le geometrie, le coordinate di geo vengono visualizzate utilizzando la precisione specificata 1-20. Se la precisione non è specificata, viene usato il valore predefinito 15.

Tipo restituito

VARCHAR

Se geo è nullo, allora viene restituito il valore nullo.

Se precision è nullo, allora viene restituito il valore nullo.

Se il risultato è una VARCHAR di dimensioni maggiori di 64 KB, viene restituito un errore.

Esempi

Il seguente comando SQL restituisce la rappresentazione esadecimale WKT di una linestring.

```
SELECT ST_AsText(ST_GeomFromText('LINESTRING(3.141592653589793
 -6.283185307179586,2.718281828459045 -1.414213562373095)', 4326));
```

```
st_astext
-------------------------------
LINESTRING(3.14159265358979 -6.28318530717959,2.71828182845905 -1.41421356237309)
```

Il seguente comando SQL restituisce la rappresentazione esadecimale WKT di una linestring. Le coordinate delle geometrie sono visualizzate con una precisione di sei cifre.

```
SELECT ST_AsText(ST_GeomFromText('LINESTRING(3.141592653589793
 -6.283185307179586,2.718281828459045 -1.414213562373095)', 4326), 6);
```

```
st_astext
---------------------------------------------
 LINESTRING(3.14159 -6.28319,2.71828 -1.41421)
```

Il seguente comando SQL restituisce la rappresentazione WKT di una geografia.

```
SELECT ST_AsText(ST_GeogFromText('LINESTRING(110 40, 2 3, -10 80, -7 9)'));
```

```
            st_astext
-----------------------------------
 LINESTRING(110 40,2 3,-10 80,-7 9)
```

ST_Azimuth

ST_Azimuth restituisce l'azimut cartesiano orientato verso il nord utilizzando le proiezioni 2D dei due punti di input.

Sintassi

```
ST_Azimuth(point1, point2)
```

Argomenti

point1

Un valore `POINT` di tipo `GEOMETRY`. L'identificatore del sistema di riferimento spaziale (SRID) di point1 deve corrispondere allo SRID di point2.

point2

Un valore `POINT` di tipo `GEOMETRY`. L'identificatore del sistema di riferimento spaziale (SRID) di point2 deve corrispondere allo SRID di point1.

Tipo restituito

Un numero che è un angolo in radianti di tipo di dati `DOUBLE PRECISION`. I valori vanno da 0 (incluso) a 2 Pi Greco (escluso).

Se point1 o point2 sono il punto vuoto, allora viene restituito un errore.

Se point1 o point2 sono nulli, allora viene restituito il valore nullo.

Se point1 e point2 sono uguali, allora viene restituito il valore nullo.

Se point1 o point2 non sono dei punti, allora viene restituito un errore.

Se point1 e point2 non hanno lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce l'azimut dei punti in input.

```
SELECT ST_Azimuth(ST_Point(1,2), ST_Point(5,6));
```

```
st_azimuth
-------------------
 0.7853981633974483
```

ST_Boundary

ST_Boundary restituisce il limite di una geometria di input nel modo seguente:

- Se la geometria di input è vuota (ovvero non contiene punti), viene restituita così com'è.
- Se la geometria di input è un punto o un multipunto non vuoto, viene restituita una raccolta di geometria vuota.
- Se l'input è una linestring o una multilinestring, viene restituito un multipunto contenente tutti i punti sul limite. Il multipunto potrebbe essere vuoto).

- Se l'input è un poligono che non ha anelli interni, viene restituita una linestring chiusa che rappresenta il suo limite.
- Se l'input è un poligono con anelli interni o multipoligono, viene restituito una multilinestring. L'anello multilinestring contiene tutti i limiti di tutti gli anelli nella geometria areale come linestring chiuse.

Per determinare l'uguaglianza dei punti, ST_Boundary opera sulla proiezione 2D della geometria di input. Se la geometria di input è vuota, una sua copia viene restituita nella stessa dimensione dell'input. Per le geometrie 3DM e 4D non vuote, le loro coordinate `m` vengono eliminate. Nel caso speciale delle multilinestring 3DZ e 4D, le coordinate `z` dei punti limite della multilinestring sono calcolate come medie dei valori z distinti dei punti limite di linestring con la stessa proiezione 2D.

Sintassi

```
ST_Boundary(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`GEOMETRY`

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è un `GEOMETRYCOLLECTION`, allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce il limite del poligono di input come una multilinestring.

```
SELECT ST_AsEWKT(ST_Boundary(ST_GeomFromText('POLYGON((0 0,10 0,10 10,0 10,0 0),(1 1,1
 2,2 1,1 1))')));
```

```
st_asewkt
```

```
--------------------
 MULTILINESTRING((0 0,10 0,10 10,0 10,0 0),(1 1,1 2,2 1,1 1))
```

ST_Buffer

ST_Buffer restituisce la geometria 2D che rappresenta tutti i punti la cui distanza dalla geometria di input proiettata sul piano cartesiano XY è minore o uguale alla distanza di input.

Sintassi

```
ST_Buffer(geom, distance)
```

```
ST_Buffer(geom, distance, number_of_segments_per_quarter_circle)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

distance

Un valore di tipo di dati DOUBLE PRECISION che rappresenta la distanza (o il raggio) del buffer.

numero_di_segmenti_per_quarto_di_ciclo

Un valore di tipo INTEGER. Questo valore determina il numero di punti per approssimare un quarto di cerchio attorno a ciascun vertice della geometria di input. Per impostazione predefinita, i valori negativi sono zero. Il valore di default è 8.

Tipo restituito

GEOMETRY

La funzione ST_Buffer restituisce una geometria bidimensionale (2D) nel piano cartesiano XY.

Se geom è un GEOMETRYCOLLECTION, allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce il buffer del linestring di input.

```
SELECT ST_AsEwkt(ST_Buffer(ST_GeomFromText('LINESTRING(1 2,5 2,5 8)'), 2));
```

```
               st_asewkt
POLYGON((-1 2,-0.96157056080646 2.39018064403226,-0.847759065022573
 2.76536686473018,-0.662939224605089 3.11114046603921,-0.414213562373093
 3.4142135623731,-0.111140466039201 3.66293922460509,0.234633135269824
 3.84775906502257,0.609819355967748 3.96157056080646,1 4,3 4,3 8,3.03842943919354
 8.39018064403226,3.15224093497743 8.76536686473018,3.33706077539491
 9.11114046603921,3.58578643762691 9.4142135623731,3.8888595339608
 9.66293922460509,4.23463313526982 9.84775906502257,4.60981935596775
 9.96157056080646,5 10,5.39018064403226 9.96157056080646,5.76536686473018
 9.84775906502257,6.11114046603921 9.66293922460509,6.4142135623731
 9.41421356237309,6.66293922460509 9.1111404660392,6.84775906502258
 8.76536686473017,6.96157056080646 8.39018064403225,7 8,7 2,6.96157056080646
 1.60981935596774,6.84775906502257 1.23463313526982,6.66293922460509
 0.888859533960796,6.41421356237309 0.585786437626905,6.1111404660392
 0.33706077539491,5.76536686473018 0.152240934977427,5.39018064403226
 0.0384294391935391,5 0,1 0,0.609819355967744 0.0384294391935391,0.234633135269821
 0.152240934977427,-0.111140466039204 0.337060775394909,-0.414213562373095
 0.585786437626905,-0.662939224605091 0.888859533960796,-0.847759065022574
 1.23463313526982,-0.961570560806461 1.60981935596774,-1 2))
```

Il seguente comando SQL restituisce il buffer della geometria del punto in input che approssima un cerchio. Poiché il comando non specifica il numero di segmenti per quarto di cerchio, per approssimare il quarto di cerchio la funzione utilizza il valore di default di otto segmenti.

```
SELECT ST_AsEwkt(ST_Buffer(ST_GeomFromText('POINT(3 4)'), 2));
```

```
               st_asewkt
POLYGON((1 4,1.03842943919354 4.39018064403226,1.15224093497743
 4.76536686473018,1.33706077539491 5.11114046603921,1.58578643762691
 5.4142135623731,1.8888595339608 5.66293922460509,2.23463313526982
 5.84775906502257,2.60981935596775 5.96157056080646,3 6,3.39018064403226
 5.96157056080646,3.76536686473019 5.84775906502257,4.11114046603921
 5.66293922460509,4.4142135623731 5.41421356237309,4.66293922460509
 5.1111404660392,4.84775906502258 4.76536686473017,4.96157056080646 4.39018064403225,5
 4,4.96157056080646 3.60981935596774,4.84775906502257 3.23463313526982,4.66293922460509
 2.8888595339608,4.41421356237309 2.58578643762691,4.1111404660392
 2.33706077539491,3.76536686473018 2.15224093497743,3.39018064403226 2.03842943919354,3
 2,2.60981935596774 2.03842943919354,2.23463313526982 2.15224093497743,1.8888595339608
 2.33706077539491,1.58578643762691 2.58578643762691,1.33706077539491
```

```
 2.8888595339608,1.15224093497743 3.23463313526982,1.03842943919354 3.60981935596774,1
 4))
```

Il seguente comando SQL restituisce il buffer della geometria del punto in input che approssima un cerchio. Poiché il comando specifica 3 come numero di segmenti per quarto di cerchio, per approssimare il quarto di cerchio la funzione utilizza tre segmenti.

```
SELECT ST_AsEwkt(ST_Buffer(ST_GeomFromText('POINT(3 4)'), 2, 3));
```

```
               st_asewkt
POLYGON((1 4,1.26794919243112 5,2 5.73205080756888,3 6,4
 5.73205080756888,4.73205080756888 5,5 4,4.73205080756888 3,4 2.26794919243112,3 2,2
 2.26794919243112,1.26794919243112 3,1 4))
```

ST_Centroid

ST_Centroid restituisce un punto che rappresenta un centroide di una geometria come segue:

- Per geometrie `POINT`, restituisce il punto le cui coordinate sono la media delle coordinate dei punti nella geometria.
- Per geometrie `LINESTRING`, restituisce il punto le cui coordinate sono la media ponderata dei punti medi dei segmenti della geometria, dove i pesi sono le lunghezze dei segmenti della geometria.
- Per geometrie `POLYGON`, restituisce il punto le cui coordinate sono la media ponderata dei centroidi di una triangolazione della geometria areale dove i pesi sono le aree dei triangoli nella triangolazione.
- Per le raccolte di geometria, restituisce la media ponderata dei centroidi delle geometrie della dimensione topologica massima nella raccolta di geometria.

Sintassi

```
ST_Centroid(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`GEOMETRY`

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce un punto centrale di una linestring di input.

```
SELECT ST_AsEWKT(ST_Centroid(ST_GeomFromText('LINESTRING(110 40, 2 3, -10 80, -7 9, -22
 -33)', 4326)))
```

```
                    st_asewkt
-----------------------------------------------------
 SRID=4326;POINT(15.6965103455214 27.0206782881905)
```

ST_Collect

ST_Collect ha due varianti. Una accetta due geometrie mentre l'altra accetta un'espressione aggregata.

La prima variante di ST_Collect crea una geometria a partire dalle geometrie in input. L'ordine delle geometrie di input viene mantenuto. Questa variante funziona come segue:

- Se entrambe le geometrie di input sono punti, allora viene restituito un `MULTIPOINT` con due punti.
- Se entrambe le geometrie di input sono linestring, allora viene restituito un `MULTILINESTRING` con due linestring.
- Se entrambe le geometrie di input sono poligoni, allora viene restituito un `MULTIPOLYGON` con due poligoni.
- In caso contrario, viene restituito un `GEOMETRYCOLLECTION` con due geometrie di input.

La seconda variante di ST_Collect crea una geometria dalle geometrie in una colonna della geometria. Non esiste un determinato ordine di restituzione delle geometrie. Specificare la clausola WITHIN GROUP (ORDER BY...) per specificare l'ordine delle geometrie restituite. Questa variante funziona come segue:

- Se tutte le righe non NULL nell'espressione di aggregazione di input sono punti, viene restituito un multipunto contenente tutti i punti nell'espressione di aggregazione.
- Se tutte le righe non NULL nell'espressione di aggregazione sono linestring, viene restituito un elemento multilinestring contenente tutte le linestring nell'espressione di aggregazione.
- Se tutte le righe non NULL nell'espressione di aggregazione sono poligoni, viene restituito un elemento multipoligono contenente tutti i poligoni nell'espressione di aggregazione.
- In caso contrario, viene restituito un `GEOMETRYCOLLECTION` contenente tutte le geometrie nell'espressione aggregata.

ST_Collect restituisce la geometria della stessa dimensione delle geometrie di input. Tutte le geometrie di input devono avere la stessa dimensione.

Sintassi

```
ST_Collect(geom1, geom2)
```

```
ST_Collect(aggregate_expression)  [WITHIN GROUP (ORDER BY sort_expression1 [ASC | DESC]
 [, sort_expression2 [ASC | DESC] ...])]
```

Argomenti

geom1

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

geom2

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

aggregate_expression

Una colonna di tipo di dati `GEOMETRY` o un'espressione che restituisce un tipo `GEOMETRY`.

[WITHIN GROUP (ORDER BY sort_expression1 [ASC | DESC] [, sort_expression2 [ASC | DESC] ...])]

Una clausola facoltativa che specifica l'ordinamento dei valori aggregati. La clausola ORDER BY contiene un elenco di espressioni di ordinamento. Le espressioni di ordinamento sono espressioni simili alle espressioni di ordinamento valide in un elenco di query select, ad esempio il nome di una colonna. È possibile specificare l'ordinamento crescente (`ASC`) o decrescente (`DESC`). Il valore predefinito è `ASC`.

Tipo restituito

GEOMETRY di sottotipo MULTIPOINT, MULTILINESTRING, MULTIPOLYGON o GEOMETRYCOLLECTION.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è il valore SRID degli oggetti geometrici in input.

Se geom1 e geom2 sono entrambi null, allora viene restituito il valore null.

Se tutte le righe di aggregate_expression sono nulle, allora viene restituito un valore nullo.

Se geom1 è null, viene restituita una copia di geom2. Allo stesso modo, se geom2 è nullo, viene restituita una copia di geom1.

Se geom1 e geom2 presentano valori SRID diversi, allora viene restituito un errore.

Se due geometrie in aggregate_expression hanno valori SRID diversi, allora viene restituito un errore.

Se la geometria restituita è maggiore della dimensione massima di un GEOMETRY, allora viene restituito un errore.

Se geom1 e geom2 hanno dimensioni diverse, allora viene restituito un errore.

Se due geometrie in aggregate_expression hanno dimensioni diverse, allora viene restituito un errore.

Esempi

Il seguente SQL restituisce una raccolta di geometrie che contiene le due geometrie di input.

```
SELECT ST_AsText(ST_Collect(ST_GeomFromText('LINESTRING(0 0,1 1)'),
 ST_GeomFromText('POLYGON((10 10,20 10,10 20,10 10))')));
```

```
st_astext
-----------
 GEOMETRYCOLLECTION(LINESTRING(0 0,1 1),POLYGON((10 10,20 10,10 20,10 10)))
```

Il seguente SQL raccoglie tutte le geometrie da una tabella in una raccolta geometrica.

```
WITH tbl(g) AS (SELECT ST_GeomFromText('POINT(1 2)', 4326) UNION ALL
SELECT ST_GeomFromText('LINESTRING(0 0,10 0)', 4326) UNION ALL
SELECT ST_GeomFromText('MULTIPOINT(13 4,8 5,4 4)', 4326) UNION ALL
```

```
SELECT NULL::geometry UNION ALL
SELECT ST_GeomFromText('POLYGON((0 0,10 0,0 10,0 0))', 4326))
SELECT ST_AsEWKT(ST_Collect(g)) FROM tbl;
```

```
st_astext
-----------
 SRID=4326;GEOMETRYCOLLECTION(POINT(1 2),LINESTRING(0 0,10 0),MULTIPOINT((13 4),(8 5),
(4 4)),POLYGON((0 0,10 0,0 10,0 0)))
```

Il seguente SQL raccoglie tutte le geometrie nella tabella raggruppate in base colonna id e ordinate in base a questo ID. In questo esempio, le geometrie risultanti vengono raggruppate per ID come segue:

- id 1: punti in un multipunto.
- id 2: linestring in una multilinestring.
- id 3: sottotipi misti in una raccolta di geometrie.
- id 4: poligoni in un multipoligono.
- id 5: null e il risultato è null.

```
WITH tbl(id, g) AS (SELECT 1, ST_GeomFromText('POINT(1 2)', 4326) UNION ALL
SELECT 1, ST_GeomFromText('POINT(4 5)', 4326) UNION ALL
SELECT 2, ST_GeomFromText('LINESTRING(0 0,10 0)', 4326) UNION ALL
SELECT 2, ST_GeomFromText('LINESTRING(10 0,20 -5)', 4326) UNION ALL
SELECT 3, ST_GeomFromText('MULTIPOINT(13 4,8 5,4 4)', 4326) UNION ALL
SELECT 3, ST_GeomFromText('MULTILINESTRING((-1 -1,-2 -2),(-3 -3,-5 -5))', 4326) UNION
 ALL
SELECT 4, ST_GeomFromText('POLYGON((0 0,10 0,0 10,0 0))', 4326) UNION ALL
SELECT 4, ST_GeomFromText('POLYGON((20 20,20 30,30 20,20 20))', 4326) UNION ALL
SELECT 1, NULL::geometry UNION ALL SELECT 2, NULL::geometry UNION ALL
SELECT 5, NULL::geometry UNION ALL SELECT 5, NULL::geometry)
SELECT id, ST_AsEWKT(ST_Collect(g)) FROM tbl GROUP BY id ORDER BY id;
```

```
 id |                                                st_asewkt

----
+-----------------------------------------------------------------------------------------
  1 | SRID=4326;MULTIPOINT((1 2),(4 5))
```

```
  2 | SRID=4326;MULTILINESTRING((0 0,10 0),(10 0,20 -5))
  3 | SRID=4326;GEOMETRYCOLLECTION(MULTIPOINT((13 4),(8 5),(4 4)),MULTILINESTRING((-1
 -1,-2 -2),(-3 -3,-5 -5)))
  4 | SRID=4326;MULTIPOLYGON(((0 0,10 0,0 10,0 0)),((20 20,20 30,30 20,20 20)))
  5 |
```

Il seguente comando SQL raccoglie tutte le geometrie da una tabella in una raccolta di geometrie. I risultati sono ordinati in ordine decrescente per `id` e quindi lessicograficamente in base alle loro coordinate x minime e massime.

```
WITH tbl(id, g) AS (
SELECT 1, ST_GeomFromText('POINT(4 5)', 4326) UNION ALL
SELECT 1, ST_GeomFromText('POINT(1 2)', 4326) UNION ALL
SELECT 2, ST_GeomFromText('LINESTRING(10 0,20 -5)', 4326) UNION ALL
SELECT 2, ST_GeomFromText('LINESTRING(0 0,10 0)', 4326) UNION ALL
SELECT 3, ST_GeomFromText('MULTIPOINT(13 4,8 5,4 4)', 4326) UNION ALL
SELECT 3, ST_GeomFromText('MULTILINESTRING((-1 -1,-2 -2),(-3 -3,-5 -5))', 4326) UNION
 ALL
SELECT 4, ST_GeomFromText('POLYGON((20 20,20 30,30 20,20 20))', 4326) UNION ALL
SELECT 4, ST_GeomFromText('POLYGON((0 0,10 0,0 10,0 0))', 4326) UNION ALL
SELECT 1, NULL::geometry UNION ALL SELECT 2, NULL::geometry UNION ALL
SELECT 5, NULL::geometry UNION ALL SELECT 5, NULL::geometry)
SELECT ST_AsEWKT(ST_Collect(g) WITHIN GROUP (ORDER BY id DESC, ST_XMin(g), ST_XMax(g)))
 FROM tbl;
```

```
                          st_asewkt

-----------------------------------------------------------------------------------------
 SRID=4326;GEOMETRYCOLLECTION(POLYGON((0 0,10 0,0 10,0 0)),POLYGON((20 20,20 30,30
 20,20 20)),MULTILINESTRING((-1 -1,-2 -2),(-3 -3,-5 -5)),MULTIPOINT((13 4),(8 5),(4
 4)),LINESTRING(0 0,10 0),LINESTRING(10 0,20 -5),POINT(1 2),POINT(4 5)
```

ST_Contains

ST_Contains restituisce true se la proiezione 2D della prima geometria di input contiene la proiezione 2D della seconda geometria di input. L'oggetto geometrico A contiene l'oggetto geometrico B se ogni punto di B è un punto di A, e le loro aree interne presentano un'intersezione non vuota.

ST_Contains(A, B) è equivalente a ST_Within(B, A).

Sintassi

```
ST_Contains(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo valore è confrontato con geom1 per determinare se è contenuto all'interno di geom1.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica se il primo poligono contiene il secondo poligono.

```
SELECT ST_Contains(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'),
 ST_GeomFromText('POLYGON((-1 3,2 1,0 -3,-1 3))'));
```

```
st_contains
-----------
 false
```

ST_ ContainsProperly

ST_ ContainsProperly restituisce true se entrambe le geometrie di input non sono vuote e tutti i punti della proiezione 2D della seconda geometria sono punti interni della proiezione 2D della prima geometria.

Sintassi

```
ST_ContainsProperly(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Il sottotipo non può essere GEOMETRYCOLLECTION.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Il sottotipo non può essere GEOMETRYCOLLECTION. Questo valore viene confrontato con geom1 per determinare se tutti i suoi punti sono punti interni di geom1.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il codice SQL seguente restituisce i valori di ST_contains e ST_ ContainsProperly dove la stringa di linea di input interseca l'interno e il limite del poligono di input (ma non il suo esterno). Il poligono contiene la linestring ma non la contiene correttamente.

```
WITH tmp(g1, g2)
```

```
AS (SELECT ST_GeomFromText('POLYGON((0 0,10 0,10 10,0 10,0 0))'),
 ST_GeomFromText('LINESTRING(5 5,10 5,10 6,5 5)')) SELECT ST_Contains(g1, g2),
 ST_ContainsProperly(g1, g2)
FROM tmp;
```

```
 st_contains | st_containsproperly
-------------+---------------------
 t           | f
```

ST_ ConvexHull

ST_ ConvexHull restituisce una geometria che rappresenta il guscio convesso dei punti non vuoti contenuti nella geometria di input.

Per l'input vuoto, la geometria risultante è la stessa della geometria di input. Per tutti gli input non vuoti, la funzione opera sulla proiezione 2D della geometria di input. Tuttavia, la dimensione della geometria di output dipende dalla dimensione della geometria di input. Più specificamente, quando la geometria di input è una geometria 3DM o 3D non vuota, le coordinate `m` vengono eliminate. In altre parole, la dimensione della geometria restituita è 2D o 3DZ, rispettivamente. Se l'input è una geometria 2D o 3DZ non vuota, la geometria risultante ha la stessa dimensione.

Sintassi

```
ST_ConvexHull(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`GEOMETRY`

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

Se geom è nullo, allora viene restituito il valore nullo.

I valori restituiti sono i seguenti.

Numero di punti sull'inviluppo convesso	Sottotipo dato di tipo geometry
0	Viene restituita una copia di geom.
1	Viene restituito un sottotipo `POINT`.
2	Viene restituito un sottotipo `LINESTRING` . I due punti della linestring restituita sono ordinati lessicograficamente.
3 o maggiore	Viene restituito un sottotipo `POLYGON` senza anelli interni. Il poligono è orientato in senso orario e il primo punto dell'anello esterno è il punto lessicograficamente più piccolo dell'anello.

Esempi

Il seguente SQL restituisce la rappresentazione estesa Well-Known Text (EWKT) di un linestring. In questo caso, l'inviluppo convesso restituito è un poligono.

```
SELECT ST_AsEWKT(ST_ConvexHull(ST_GeomFromText('LINESTRING(0 0,1 0,0 1,1 1,0.5 0.5)')))
 as output;
```

```
output
-------------
POLYGON((0 0,0 1,1 1,1 0,0 0))
```

Il seguente comando SQL restituisce la rappresentazione in formato EWKT di una linestring. In questo caso, l'inviluppo convesso restituito è una linestring.

```
SELECT ST_AsEWKT(ST_ConvexHull(ST_GeomFromText('LINESTRING(0 0,1 1,0.2 0.2,0.6 0.6,0.5
 0.5)'))) as output;
```

```
output
-------------
LINESTRING(0 0,1 1)
```

Il seguente SQL restituisce la rappresentazione EWKT di un multipunto. In questo caso, l'inviluppo convesso restituito è un punto.

```
SELECT ST_AsEWKT(ST_ConvexHull(ST_GeomFromText('MULTIPOINT(0 0,0 0,0 0)'))) as output;
```

```
output
-------------
POINT(0 0)
```

ST_ CoveredBy

ST_ CoveredBy restituisce true se la proiezione 2D della prima geometria di input è coperta dalla proiezione 2D della seconda geometria di input. L'oggetto geometrico A è ricoperta dall'oggetto geometricoB se entrambe sono non vuote e ogni punto in A è un punto in B.

ST_ CoveredBy (A,) è equivalente a ST_covers (,B). B A

Sintassi

```
ST_CoveredBy(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo valore è confrontato con geom2 per determinare se è ricoperto da geom2.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica se il primo poligono è ricoperto dal secondo poligono.

```
SELECT ST_CoveredBy(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'),
 ST_GeomFromText('POLYGON((-1 3,2 1,0 -3,-1 3))'));
```

```
st_coveredby
------------
 true
```

ST_Covers

ST_Covers restituisce true se la proiezione 2D della prima geometria di input copre la proiezione 2D della seconda geometria di input. L'oggetto geometrico `A` ricopre l'oggetto geometrico `B` se entrambe sono non vuote e ogni punto in `B` è un punto in `A`.

ST_covers (`A`,`B`) è equivalente a ST_ (,). CoveredBy `B` `A`

Sintassi

```
ST_Covers(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

geom2

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo valore è confrontato con geom1 per determinare se è ricoperto da geom1.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica se il primo poligono ricopre il secondo poligono.

```
SELECT ST_Covers(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'),
 ST_GeomFromText('POLYGON((-1 3,2 1,0 -3,-1 3))'));
```

```
st_covers
-----------
 false
```

ST_Crosses

ST_Crosses restituisce true se le proiezioni 2D delle due geometrie di input si incrociano tra loro.

Sintassi

```
ST_Crosses(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

`BOOLEAN`

Se geom1 o geom2 sono nulli, allora viene restituito un errore.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica se il primo poligono interseca il secondo multipunto. In questo esempio, il multipunto interseca sia l'interno che l'esterno del poligono, motivo per cui ST_Crosses restituisce true.

```
SELECT ST_Crosses (ST_GeomFromText('polygon((0 0,10 0,10 10,0 10,0 0))'),
 ST_GeomFromText('multipoint(5 5,0 0,-1 -1)'));
```

```
st_crosses
-------------
 true
```

Il seguente comando SQL verifica se il primo poligono interseca il secondo multipunto. In questo esempio, il multipunto interseca l'esterno del poligono ma non l'interno, motivo per cui ST_Crosses restituisce false.

```
SELECT ST_Crosses (ST_GeomFromText('polygon((0 0,10 0,10 10,0 10,0 0))'),
 ST_GeomFromText('multipoint(0 0,-1 -1)'));
```

```
st_crosses
-------------
 false
```

ST_Dimension

ST_Dimension restituisce la dimensione intrinseca di un oggetto geometrico in input. La dimensione intrinseca è il valore della dimensione del sottotipo che è definito in un oggetto geometrico.

Per gli input di geometria 3DM, 3DZ e 4D, ST_Dimension restituisce lo stesso risultato degli input geometrici 2D.

Sintassi

```
ST_Dimension(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

INTEGER rappresentante la dimensione intrinseca di geom.

Se geom è nullo, allora viene restituito il valore nullo.

I valori restituiti sono i seguenti.

Valore restituito	Sottotipo dato di tipo geometry
0	Restituito se geom è di sottotipo POINT o MULTIPOINT
1	Restituito se geom è di sottotipo LINESTRING o MULTILINESTRING
2	Restituito se geom è di sottotipo POLYGON o MULTIPOLYGON
0	Restituito se geom è di sottotipo GEOMETRYCOLLECTION
La dimensione maggiore dei componenti della raccolta	Restituito se geom è di sottotipo GEOMETRYCOLLECTION

Esempi

Il seguente comando SQL converte una rappresentazione Well-Known Text (WKT) di un oggetto LINESTRING a quattro punti in un oggetto GEOMETRY e restituisce la dimensione della linestring.

```
SELECT ST_Dimension(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27
 29.31,77.29 29.07)'));
```

```
st_dimension
-------------
1
```

ST_Disjoint

ST_Disjoint restituisce il valore true se le proiezioni 2D delle due geometrie in input non hanno punti in comune.

Sintassi

```
ST_Disjoint(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica se il primo poligono è disgiunto dal secondo poligono.

```
SELECT ST_Disjoint(ST_GeomFromText('POLYGON((0 0,10 0,10 10,0 10,0 0),(2 2,2 5,5 5,5
 2,2 2))'), ST_Point(4, 4));
```

```
st_disjoint
-----------
 true
```

ST_Distance

Per le geometrie di input, ST_Distance restituisce la distanza euclidea minima tra le proiezioni 2D dei due valori della geometria di input.

Per geometrie 3DM, 3DZ, 4D, ST_Distance restituisce la distanza euclidea tra le proiezioni 2D dei due valori della geometria di input.

Per le aree geografiche di input, ST_Distance restituisce la distanza geodetica dei due punti 2D. L'unità di distanza è espressa in metri. Per aree geografiche diverse da punti e punti vuoti viene restituito un errore.

Sintassi

```
ST_Distance(geo1, geo2)
```

Argomenti

geo1

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY. Il tipo di dati di geo1 deve corrispondere esattamente a geo2.

geo2

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY. Il tipo di dati di geo2 deve corrispondere esattamente a geo1.

Tipo restituito

`DOUBLE PRECISION` nelle stesse unità delle geometrie o geografie in input.

Se geo1 o geo2 è nullo o vuoto, allora viene restituito il valore nullo.

Se geo1 e geo2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito un errore.

Se geo1 o geo2 è una collezione di geometrie, allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce la distanza tra i due poligoni.

```
SELECT ST_Distance(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'),
 ST_GeomFromText('POLYGON((-1 -3,-2 -1,0 -3,-1 -3))'));
```

```
  st_distance
-----------
1.4142135623731
```

La seguente istruzione SQL restituisce la distanza (in metri) tra la Porta di Brandeburgo e l'edificio del Reichstag a Berlino utilizzando un tipo di dati GEOGRAPHY.

```
SELECT ST_Distance(ST_GeogFromText('POINT(13.37761826722198 52.516411678282445)'),
 ST_GeogFromText('POINT(13.377950831464005 52.51705102546893)'));
```

```
   st_distance
------------------
 74.64129172609631
```

ST_ DistanceSphere

ST_ DistanceSphere restituisce la distanza tra due geometrie di punti che si trovano su una sfera.

Sintassi

```
ST_DistanceSphere(geom1, geom2)
```

```
ST_DistanceSphere(geom1, geom2, radius)
```

Argomenti

geom1

Un valore puntuale in gradi di tipo di dati GEOMETRY disposti su una sfera. La prima coordinata del punto è il valore della longitudine. La seconda coordinata del punto è il valore della latitudine. Per le geometrie 3DZ, 3DM o 4D, vengono utilizzate solo le prime due coordinate.

geom2

Un valore puntuale in gradi di tipo di dati GEOMETRY disposti su una sfera. La prima coordinata del punto è il valore della longitudine. La seconda coordinata del punto è il valore della latitudine. Per le geometrie 3DZ, 3DM o 4D, vengono utilizzate solo le prime due coordinate.

raggio

Il raggio di una sfera di tipo di dato DOUBLE PRECISION. Se non è indicato un raggio, la sfera considerata per impostazione predefinita è la Terra e il raggio è calcolato in base alla rappresentazione World Geodetic System (WGS) 84 dell'ellissoide.

Tipo restituito

DOUBLE PRECISION nelle stesse unità del raggio. Se non viene fornito alcun raggio, la distanza è in metri.

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se non viene indicato un raggio, allora il risultato è in metri misurati sulla superficie della Terra.

Se il raggio è un numero negativo, viene restituito un errore.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 non sono dei punti, allora viene restituito un errore.

Esempi

La seguente istruzione SQL di esempio calcola la distanza in chilometri tra due punti sulla Terra.

```
SELECT ROUND(ST_DistanceSphere(ST_Point(-122, 47), ST_Point(-122.1, 47.1))/ 1000, 0);
```

```
  round
-----------
 13
```

Il seguente comando SQL di esempio calcola le distanze in chilometri tra tre località aeroportuali in Germania: Berlin Tegel (TXL), Munich International (MUC) e Frankfurt International (FRA).

```
WITH airports_raw(code,lon,lat) AS (
(SELECT 'MUC', 11.786111, 48.353889) UNION
(SELECT 'FRA', 8.570556, 50.033333) UNION
(SELECT 'TXL', 13.287778, 52.559722)),
airports1(code,location) AS (SELECT code, ST_Point(lon, lat) FROM airports_raw),
airports2(code,location) AS (SELECT * from airports1)
SELECT (airports1.code || ' <-> ' || airports2.code) AS airports,
round(ST_DistanceSphere(airports1.location, airports2.location) / 1000, 0) AS
 distance_in_km
FROM airports1, airports2 WHERE airports1.code < airports2.code ORDER BY 1;
```

```
  airports   | distance_in_km
-------------+----------------
 FRA <-> MUC |            299
 FRA <-> TXL |            432
 MUC <-> TXL |            480
```

ST_DWithin

ST_DWithin restituisce true se la distanza euclidea tra le proiezioni 2D dei due valori della geometria di input non è maggiore di un valore soglia.

Sintassi

```
ST_DWithin(geom1, geom2, threshold)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

soglia

Un valore di tipo DOUBLE PRECISION. Questo valore è espresso nelle unità di misura utilizzate per gli argomenti di input.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se la soglia è negativa, allora viene restituito un errore.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL controlla se la distanza tra due poligoni è inferiore a cinque unità.

```
SELECT ST_DWithin(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'),
 ST_GeomFromText('POLYGON((-1 3,2 1,0 -3,-1 3))'),5);
```

```
st_dwithin
-----------
 true
```

ST_ EndPoint

ST_ EndPoint restituisce l'ultimo punto di una stringa di linee di input. Il valore dell'identificatore del sistema di riferimento spaziale (SRID) del risultato è lo stesso della geometria di input. La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_EndPoint(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere LINESTRING.

Tipo restituito

GEOMETRY

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom non è un LINESTRING, viene restituito il valore nullo.

Esempi

Il seguente codice SQL restituisce una rappresentazione estesa di testo noto (EWKT) di una LINESTRING a quattro punti a un oggetto GEOMETRY e restituisce il punto finale della linestring.

```
SELECT ST_AsEWKT(ST_EndPoint(ST_GeomFromText('LINESTRING(0 0,10 0,10 10,5 5,0
 5)',4326)));
```

```
st_asewkt
-------------
 SRID=4326;POINT(0 5)
```

ST_Envelope

ST_Envelope restituisce il riquadro di delimitazione minimo della geometria di input, come segue:

- Se la geometria di input è vuota, la geometria restituita è una copia della geometria di input.
- Se il riquadro di delimitazione minimo della geometria di input degenera in un punto, la geometria restituita è un punto.

- Se il riquadro di delimitazione minimo della geometria di input è unidimensionale, viene restituita una linea a due punti.
- Se nessuna delle precedenti condizioni è vera, la funzione restituisce un poligono orientato in senso orario i cui vertici sono gli angoli del riquadro di delimitazione minimo.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è il valore SRID degli oggetti geometrici in input.

Per tutti gli input non vuoti, la funzione opera sulla proiezione 2D della geometria di input.

Sintassi

```
ST_Envelope(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

GEOMETRY

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente codice SQL converte una rappresentazione di testo noto (WKT) di una LINESTRING a quattro punti in un oggetto GEOMETRY e restituisce un poligono i cui vertici e angoli sono il riquadro di delimitazione minimo.

```
SELECT ST_AsText(ST_Envelope(ST_GeomFromText('GEOMETRYCOLLECTION(POLYGON((0 0,10 0,0
 10,0 0)),LINESTRING(20 10,20 0,10 0))')));
```

```
    st_astext
-----------------------------------
  POLYGON((0 0,0 10,20 10,20 0,0 0))
```

ST_Equals

ST_Equals restituisce il valore true se le proiezioni 2D delle geometrie di input sono geometricamente uguali. Le geometrie sono considerate geometricamente uguali se hanno set di punti uguali e le loro aree interne presentano un'intersezione non vuota.

Sintassi

```
ST_Equals(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo valore è confrontato con geom1 per determinare se è uguale a geom1.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito un errore.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica se i due poligoni sono geometricamente uguali.

```
SELECT ST_Equals(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'),
 ST_GeomFromText('POLYGON((-1 3,2 1,0 -3,-1 3))'));
```

```
st_equals
```

```
-----------
 false
```

Il seguente comando SQL verifica se le due linee sono geometricamente uguali.

```
SELECT ST_Equals(ST_GeomFromText('LINESTRING(1 0,10 0)'), ST_GeomFromText('LINESTRING(1
 0,5 0,10 0)'));
```

```
st_equals
-----------
 true
```

ST_ ExteriorRing

ST_ ExteriorRing restituisce una stringa a linee chiuse che rappresenta l'anello esterno di un poligono di input. La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_ExteriorRing(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`GEOMETRY` di sottotipo `LINESTRING`.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom non è un poligono, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituito il poligono vuoto.

Esempi

Il seguente SQL restituisce l'anello esterno di un poligono come linestring.

```
SELECT ST_AsEWKT(ST_ExteriorRing(ST_GeomFromText('POLYGON((7 9,8 7,11 6,15 8,16 6,17
 7,17 10,18 12,17 14,15 15,11 15,10 13,9 12,7 9),(9 9,10 10,11 11,11 10,10 8,9 9),(12
 14,15 14,13 11,12 14))')));
```

```
st_asewkt
-----------
 LINESTRING(7 9,8 7,11 6,15 8,16 6,17 7,17 10,18 12,17 14,15 15,11 15,10 13,9 12,7 9)
```

ST_Force2D

ST_Force2D restituisce una geometria 2D della geometria di input. Per le geometrie 2D, viene restituita una copia dell'input. Per le geometrie 3DZ, 3DM e 4D, ST_Force2D proietta la geometria sul piano XY cartesiano. I punti vuoti nella geometria di input rimangono punti vuoti nella geometria di output.

Sintassi

```
ST_Force2D(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`GEOMETRY`.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituita la geometria vuota.

Esempi

Il seguente SQL restituisce una geometria 2D da una geometria 3DZ.

```
SELECT ST_AsEWKT(ST_Force2D(ST_GeomFromText('MULTIPOINT Z(0 1 2, EMPTY, 2 3 4, 5 6
 7)')));
```

```
st_asewkt
-----------
  MULTIPOINT((0 1),EMPTY,(2 3),(5 6))
```

ST_Force3D

ST_Force3D è un alias di ST_Force3DZ. Per ulteriori informazioni, consultare ST_Force3DZ.

ST_Force3DM

ST_Force3DM restituisce una geometria 3DM della geometria di input. Per le geometrie 2D, le coordinate `m` dei punti non vuoti nella geometria di output sono tutte impostate su `0`. Per geometrie 3DM, viene restituita una copia della geometria di input. Per le geometrie 3DZ, la geometria viene proiettata sul piano cartesiano XY, e le coordinate `m` dei punti non vuoti nella geometria di output sono tutte impostate su `0`. Per le geometrie 4D, la geometria viene proiettata nello spazio cartesiano XYM. I punti vuoti nella geometria di input rimangono punti vuoti nella geometria di output.

Sintassi

```
ST_Force3DM(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`GEOMETRY`.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituita la geometria vuota.

Esempi

Il seguente SQL restituisce una geometria 3DM da una geometria 3DZ.

```
SELECT ST_AsEWKT(ST_Force3DM(ST_GeomFromText('MULTIPOINT Z(0 1 2, EMPTY, 2 3 4, 5 6
 7)')));
```

```
st_asewkt
-----------
  MULTIPOINT M ((0 1 0),EMPTY,(2 3 0),(5 6 0))
```

ST_Force3DZ

ST_Force3DZ restituisce una geometria 3DZ della geometria di input. Per le geometrie 2D, le coordinate z dei punti non vuoti nella geometria di output sono tutte impostate su `0`. Per le geometrie 3DM, la geometria viene proiettata sul piano XY cartesiano, e le coordinate z dei punti non vuoti nella geometria di output sono tutte impostate su `0`. Per geometrie 3DZ, viene restituita una copia della geometria di input. Per le geometrie 4D, la geometria viene proiettata nello spazio cartesiano XYZ. I punti vuoti nella geometria di input rimangono punti vuoti nella geometria di output.

Sintassi

```
ST_Force3DZ(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`GEOMETRY`.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituita la geometria vuota.

Esempi

Il seguente SQL restituisce una geometria 3DZ da una geometria 3DM.

```
SELECT ST_AsEWKT(ST_Force3DZ(ST_GeomFromText('MULTIPOINT M(0 1 2, EMPTY, 2 3 4, 5 6
 7)')));
```

```
st_asewkt
-----------
  MULTIPOINT Z ((0 1 0),EMPTY,(2 3 0),(5 6 0))
```

ST_Force4D

ST_Force4D restituisce una geometria 4D della geometria di input. Per le geometrie 2D, le coordinate `z` e `m` dei punti non vuoti nella geometria di output sono tutte impostate su `0`. Per le geometrie 3DM, le coordinate `z` dei punti non vuoti nella geometria di output sono tutte impostate su `0`. Per le geometrie 3DZ, le coordinate `m` dei punti non vuoti nella geometria di output sono tutte impostate su `0`. Per geometrie 4D, viene restituita una copia della geometria di input. I punti vuoti nella geometria di input rimangono punti vuoti nella geometria di output.

Sintassi

```
ST_Force4D(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`GEOMETRY`.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituita la geometria vuota.

Esempi

Il seguente SQL restituisce una geometria 4D da una geometria 3DM.

```
SELECT ST_AsEWKT(ST_Force4D(ST_GeomFromText('MULTIPOINT M(0 1 2, EMPTY, 2 3 4, 5 6
 7)')));
```

```
st_asewkt
-----------
  MULTIPOINT ZM ((0 1 0 2),EMPTY,(2 3 0 4),(5 6 0 7))
```

ST_ GeoHash

ST_ GeoHash restituisce la `geohash` rappresentazione del punto di input con la precisione specificata. Il valore di precisione di default è 20. Per ulteriori informazioni sulla definizione di geohash, consulta Geohash su Wikipedia.

Sintassi

```
ST_GeoHash(geom)
```

```
ST_GeoHash(geom, precision)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

precisione

Un valore di tipo `INTEGER`. Il valore di default è 20.

Tipo restituito

GEOMETRY

La funzione restituisce la rappresentazione geohash del punto di input.

Se il punto di input è vuoto, la funzione restituisce null.

Se la geometria di input non è un punto, la funzione restituisce un errore.

Esempi

Il seguente SQL restituisce la rappresentazione geohash del punto di input.

```
SELECT ST_GeoHash(ST_GeomFromText('POINT(45 -45)'), 25) AS geohash;
```

```
          geohash
---------------------------
 m0000000000000000000gzz
```

Il seguente comando SQL restituisce null perché il punto di input è vuoto.

```
SELECT ST_GeoHash(ST_GeomFromText('POINT EMPTY'), 10) IS NULL AS result;
```

```
 result
---------
 true
```

ST_ GeogFromText

ST_ GeogFromText costruisce un oggetto geografico a partire da una rappresentazione in testo noto (WKT) o testo esteso noto (EWKT) di una geografia di input.

Sintassi

```
ST_GeogFromText(wkt_string)
```

Argomenti

wkt_string

Un valore di tipo VARCHAR che è una rappresentazione in formato WKT o EWKT di un oggetto geografico.

Tipo restituito

GEOGRAPHY

Se il valore SRID è impostato sul valore fornito nell'input. Se SRID non è fornito, viene impostato su 4326.

Se wkt_string è nullo, allora viene restituito nullo.

Se wkt_string non è valido, allora viene restituito un errore.

Esempi

Il seguente comando SQL costruisce un poligono a partire da un oggetto geografico con un valore SRID.

```
SELECT ST_AsEWKT(ST_GeogFromText('SRID=4324;POLYGON((0 0,0 1,1 1,10 10,1 0,0 0))'));
```

```
  st_asewkt
-----------------------------------------------
 SRID=4324;POLYGON((0 0,0 1,1 1,10 10,1 0,0 0))
```

Il seguente comando SQL costruisce un poligono a partire da un oggetto geografico. Il valore SRID è impostato su 4326.

```
SELECT ST_AsEWKT(ST_GeogFromText('POLYGON((0 0,0 1,1 1,10 10,1 0,0 0))'));
```

```
 st_asewkt
-----------------------------------------------
```

```
 SRID=4326;POLYGON((0 0,0 1,1 1,10 10,1 0,0 0))
```

GeogFromST_ WKB

ST_ GeogFrom WKB costruisce un oggetto geografico a partire da una rappresentazione binaria esadecimale nota (WKB) di una geografia di input.

Sintassi

```
ST_GeogFromWKB(wkb_string)
```

Argomenti

wkb_string

Un valore di tipo VARCHAR che è una rappresentazione esadecimale in formato WKB di un oggetto geografico.

Tipo restituito

GEOGRAPHY

Se viene fornito il valore SRID, viene impostato sul valore fornito. Se SRID non è fornito, viene impostato su 4326.

Se wkb_string è nullo, allora viene restituito il valore nullo.

Se wkb_string non è valido, allora viene restituito un errore.

Esempi

Il seguente comando SQL costruisce una geografia a partire da un valore WKB esadecimale.

```
SELECT
 ST_AsEWKT(ST_GeogFromWKB('0103000000010000000500000000000000000000000000000000000000000000000000000
```

```
 st_asewkt
------------------------------------------
 SRID=4326;POLYGON((0 0,0 1,1 1,1 0,0 0))
```

ST_GeometryN

ST_Geometryn restituisce una geometria puntata dall'indice di input della geometria di input, come segue:

- Se l'input è un punto, una linestring o un poligono, viene restituita una geometria come se l'indice fosse uguale a uno (1) e nullo se l'indice è diverso da uno (1).
- Se l'input è un multipunto, una multilinestring, un multipoligono oppure una raccolta di geometria, allora il punto, la linestring, il poligono o la geometria viene restituita come indicato da un indice di input.

L'indice è basato su uno. L'identificatore del sistema di riferimento spaziale (SRID) del risultato è lo stesso della geometria di input. La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_GeometryN(geom, index)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

indice

Valore del tipo di dati INTEGER che rappresenta la posizione di un indice basato su un uno.

Tipo restituito

GEOMETRY

Se geom o index sono nulli, allora viene restituito il valore nullo.

Se index è fuori intervallo, viene restituito un errore.

Esempi

Il seguente codice SQL restituisce le geometrie in una raccolta di geometrie.

```
WITH tmp1(idx) AS (SELECT 1 UNION SELECT 2),
```

```
tmp2(g) AS (SELECT ST_GeomFromText('GEOMETRYCOLLECTION(POLYGON((0 0,10 0,0 10,0
 0)),LINESTRING(20 10,20 0,10 0))'))
SELECT idx, ST_AsEWKT(ST_GeometryN(g, idx)) FROM tmp1, tmp2 ORDER BY idx;
```

```
 idx |          st_asewkt
-----+-----------------------------
   1 | POLYGON((0 0,10 0,0 10,0 0))
   2 | LINESTRING(20 10,20 0,10 0)
```

ST_ GeometryType

ST_ GeometryType restituisce il sottotipo di una geometria di input come stringa.

Per gli input geometrici 3DM, 3DZ e 4D, GeometryType ST_ restituisce lo stesso risultato degli input geometrici 2D.

Sintassi

```
ST_GeometryType(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`VARCHAR` che rappresenta il sottotipo di geom.

Se geom è nullo, allora viene restituito il valore nullo.

I valori restituiti sono i seguenti.

Valore stringa restituito	Sottotipo dato di tipo geometry
`ST_Point`	Restituito se geom è di sottotipo `POINT`
`ST_LineString`	Restituito se geom è di sottotipo `LINESTRING`

Valore stringa restituito	Sottotipo dato di tipo geometry
ST_Polygon	Restituito se geom è di sottotipo POLYGON
ST_MultiPoint	Restituito se geom è di sottotipo MULTIPOINT
ST_MultiLineString	Restituito se geom è di sottotipo MULTILINESTRING
ST_MultiPolygon	Restituito se geom è di sottotipo MULTIPOLYGON
ST_GeometryCollection	Restituito se geom è di sottotipo GEOMETRYCOLLECTION

Esempi

Il seguente comando SQL restituisce il sottotipo dell'oggetto geometrico linestring in input.

```
SELECT ST_GeometryType(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27
 29.31,77.29 29.07)'));
```

```
st_geometrytype
-------------
 ST_LineString
```

ST_ EWKB GeomFrom

ST_ GeomFrom EWKB costruisce un oggetto geometrico a partire dalla rappresentazione binaria estesa ben nota (EWKB) di una geometria di input.

ST_ GeomFrom EWKB accetta geometrie 3DZ, 3DM e 4D scritte in formato esadecimale WKB ed EWKB.

Sintassi

```
ST_GeomFromEWKB(ewkb_string)
```

Argomenti

ewkb_string

Un valore di tipo VARCHAR che è una rappresentazione esadecimale in formato EWKB di un oggetto geometrico.

Tipo restituito

GEOMETRY

Se ewkb_string è nullo, allora viene restituito il valore nullo.

Se ewkb_string non è valido, allora viene restituito un errore.

Esempi

Il seguente comando SQL costruisce un poligono a partire da un valore EWKB e restituisce la rappresentazione EWKT di un poligono.

```
SELECT
 ST_AsEWKT(ST_GeomFromEWKB('0103000020E6100000010000000500000000000000000000000000000000000000000
```

```
 st_asewkt
-------------------------------
 SRID=4326;POLYGON((0 0,0 1,1 1,1 0,0 0))
```

ST_ GeomFrom EWKT

ST_ GeomFrom EWKT costruisce un oggetto geometrico a partire dalla rappresentazione in formato extended known text (EWKT) di una geometria di input.

ST_ GeomFrom EWKT accetta 3DZ, 3DM e 4D dove il tipo di geometria ha il prefisso Z, M o ZM, rispettivamente.

Sintassi

```
ST_GeomFromEWKT(ewkt_string)
```

Argomenti

ewkt_string

Un valore di tipo di dati VARCHAR o un'espressione che restituisce un tipo VARCHAR, ovvero una rappresentazione in formato EWKT di una geometria.

È possibile utilizzare la parola chiave WKT EMPTY per designare un punto vuoto, un multipunto con un punto vuoto o una raccolta di geometria con un punto vuoto. Nell'esempio seguente viene creato un punto vuoto.

```
ST_GeomFromEWKT('SRID=4326;POINT EMPTY');
```

Tipo restituito

GEOMETRY

Se ewkt_string è null, allora viene restituito null.

Se ewkt_string non è valido, allora viene restituito un errore.

Esempi

Il seguente comando SQL costruisce un multilinestring a partire da un valore EWKT e restituisce una geometria. Restituisce inoltre il risultato ST_AsEWKT della geometria.

```
SELECT ST_GeomFromEWKT('SRID=4326;MULTILINESTRING((1 0,1 0),(2 0,3 0),(4 0,5 0,6 0))')
 as geom, ST_AsEWKT(geom);
```

```
                                                    geom

                                          |                        st_asewkt

-------------------------------------------------------------------------------------------
+----------------------------------------------------------
 0105000020E6100000030000000102000000020000000000000000F03F000000000000000000000000000000F03F00
 | SRID=4326;MULTILINESTRING((1 0,1 0),(2 0,3 0),(4 0,5 0,6 0))
```

ST_ GeomFromGeoHash

ST_ GeomFromGeoHash costruisce un oggetto geometrico a partire dalla rappresentazione geohash di una geometria di input. ST_ GeomFromGeoHash restituisce una geometria bidimensionale (2D) con l'identificatore di riferimento spaziale (SRID) pari a zero (0). Per ulteriori informazioni sul formato geohash, consulta la voce Geohash su Wikipedia.

Sintassi

```
ST_GeomFromGeoHash(geohash_string)
```

```
ST_GeomFromGeoHash(geohash_string, precision)
```

Argomenti

geohash_string

Un valore del tipo di dati VARCHAR o un'espressione che restituisce un tipo VARCHAR, ovvero una rappresentazione in formato geohash di una geometria.

precisione

Un valore del tipo di dati INTEGER che rappresenta il livello di precisione del geohash. Il valore è il numero di caratteri del geohash da utilizzare come livello di precisione. Se il valore non è specificato, è minore di zero o è maggiore della lunghezza di geohash_string, viene utilizzata la lunghezza di geohash_string.

Tipo restituito

GEOMETRY

Se il valore di geojson_string è null, viene restituito un valore nullo.

Se il valore di geojson_string non è valido, viene restituito un errore.

Esempi

Il seguente SQL restituisce un poligono con precisione elevata.

```
SELECT ST_AsText(ST_GeomFromGeoHash('9qqj7nmxncgyy4d0dbxqz0'));
```

```
 st_asewkt
-----------------------
 POLYGON((-115.172816 36.114646,-115.172816 36.114646,-115.172816 36.114646,-115.172816
 36.114646,-115.172816 36.114646))
```

Il seguente SQL restituisce un punto con precisione elevata.

```
SELECT ST_AsText(ST_GeomFromGeoHash('9qqj7nmxncgyy4d0dbxqz00'));
```

```
 st_asewkt
-----------------------
 POINT(-115.172816 36.114646)
```

Il seguente SQL restituisce un poligono con precisione bassa.

```
SELECT ST_AsText(ST_GeomFromGeoHash('9qq'));
```

```
 st_asewkt
-----------------------
 POLYGON((-115.3125 35.15625,-115.3125 36.5625,-113.90625 36.5625,-113.90625
 35.15625,-115.3125 35.15625))
```

Il seguente SQL restituisce un poligono con precisione 3.

```
SELECT ST_AsText(ST_GeomFromGeoHash('9qqj7nmxncgyy4d0dbxqz0', 3));
```

```
 st_asewkt
-----------------------
 POLYGON((-115.3125 35.15625,-115.3125 36.5625,-113.90625 36.5625,-113.90625
 35.15625,-115.3125 35.15625))
```

GeomFromGeoST_ JSON

ST_ GeomFromGeo JSON costruisce un oggetto geometrico dalla rappresentazione GeoJSON di una geometria di input. Per ulteriori informazioni sul formato GeoJSON, consulta la voce GeoJSON su Wikipedia.

Se è presente almeno un punto con tre o più coordinate, la geometria risultante è 3DZ, dove la componente Z è zero per i punti che hanno solo due coordinate. Se tutti i punti nell'input GeoJSON contengono due coordinate o sono vuoti, GeomFromGeo ST_ JSON restituisce una geometria 2D. La geometria restituita include sempre l'identificatore del sistema di riferimento spaziale (SRID) impostato su 4326.

Sintassi

```
ST_GeomFromGeoJSON(geojson_string)
```

Argomenti

geojson_string

Un valore con tipo di dati VARCHAR o un'espressione che restituisce un tipo VARCHAR, ovvero una rappresentazione in formato GeoJSON di una geometria.

Tipo restituito

GEOMETRY

Se geojson_string è nullo, viene restituito un valore nullo.

Se geojson_string non è valido, viene restituito un errore.

Esempi

La seguente istruzione SQL restituisce una geometria 2D rappresentata nell'input GeoJSON.

```
SELECT ST_AsEWKT(ST_GeomFromGeoJSON('{"type":"Point","coordinates":[1,2]}'));
```

```
 st_asewkt
```

```
-----------------------
 SRID=4326;POINT(1 2)
```

La seguente istruzione SQL restituisce una geometria 3DZ rappresentata nell'input GeoJSON.

```
SELECT ST_AsEWKT(ST_GeomFromGeoJSON('{"type":"LineString","coordinates":[[1,2,3],
[4,5,6],[7,8,9]]}'));
```

```
 st_asewkt
-----------------------------------------
 SRID=4326;LINESTRING Z (1 2 3,4 5 6,7 8 9)
```

La seguente istruzione SQL restituisce una geometria 3DZ quando solo un punto ha tre coordinate mentre tutti gli altri punti hanno due coordinate nel formato GeoJSON di input.

```
SELECT ST_AsEWKT(ST_GeomFromGeoJSON('{"type":"Polygon","coordinates":[[[0, 0],[0, 1,
 8],[1, 0],[0, 0]]]}'));
```

```
 st_asewkt
-----------------------------------------------
 SRID=4326;POLYGON Z ((0 0 0,0 1 8,1 0 0,0 0 0))
```

ST_ GeomFromGeoSquare

ST_ GeomFromGeoSquare restituisce una geometria che copre l'area rappresentata da un valore geosquare di input. La geometria restituita è sempre bidimensionale. Per calcolare un valore geosquare, consulta ST_ GeoSquare.

Sintassi

```
ST_GeomFromGeoSquare(geosquare)
```

```
ST_GeomFromGeoSquare(geosquare, max_depth)
```

Argomenti

geosquare

Un valore di tipo di dati BIGINT o un'espressione che restituisce un tipo BIGINT che è un valore geosquare che descrive la sequenza di suddivisioni effettuata sul dominio iniziale per raggiungere il quadrato desiderato. Questo valore è calcolato da ST_ GeoSquare.

max_depth

Un valore di tipo di dati INTEGER che rappresenta il numero massimo di suddivisioni di dominio create nel dominio iniziale. Il valore deve essere maggiore o uguale a 1.

Tipo restituito

GEOMETRY

Se geosquare non è valido, la funzione restituisce un errore.

Se l'input max_depth non rientra nell'intervallo, la funzione restituisce un errore.

Esempi

Il seguente comando SQL restituisce una geometria da un valore geosquare.

```
SELECT ST_AsText(ST_GeomFromGeoSquare(797852));
```

```
 st_astext
-----------------------------------------------------------------------------------------------
 POLYGON((13.359375 52.3828125,13.359375 52.734375,13.7109375 52.734375,13.7109375
 52.3828125,13.359375 52.3828125))
```

Il seguente comando SQL restituisce una geometria da un valore geosquare e una profondità massima di 3.

```
SELECT ST_AsText(ST_GeomFromGeoSquare(797852, 3));
```

```
 st_astext
-------------------------------------
 POLYGON((0 45,0 90,45 90,45 45,0 45))
```

Il seguente comando SQL calcola innanzitutto il valore geosquare per Seattle specificando la coordinata x come longitudine e la coordinata y come latitudine (-122,3, 47,6). Quindi restituisce il poligono per geosquare. Sebbene l'output sia una geometria bidimensionale, può essere utilizzata per calcolare dati spaziali in termini di longitudine e latitudine.

```
SELECT ST_AsText(ST_GeomFromGeoSquare(ST_GeoSquare(ST_Point(-122.3, 47.6))));
```

```
 st_astext
-----------------------------------------------------------------------------------------------------------
POLYGON((-122.335167014971 47.6080129947513,-122.335167014971
 47.6080130785704,-122.335166931152 47.6080130785704,-122.335166931152
 47.6080129947513,-122.335167014971 47.6080129947513))
```

ST_ GeomFromText

ST_ GeomFromText costruisce un oggetto geometrico a partire da una rappresentazione testuale ben nota (WKT) di una geometria di input.

ST_ GeomFromText accetta 3DZ, 3DM e 4D dove il tipo di geometria ha il prefisso Z, M o ZM, rispettivamente.

Sintassi

```
ST_GeomFromText(wkt_string)
```

```
ST_GeomFromText(wkt_string, srid)
```

Argomenti

wkt_string

Un valore di tipo VARCHAR che è una rappresentazione in formato WKT di un oggetto geometrico.

È possibile utilizzare la parola chiave WKT EMPTY per designare un punto vuoto, un multipunto con un punto vuoto o una raccolta di geometria con un punto vuoto. L'esempio seguente crea un punto multiplo con un punto vuoto e uno non vuoto.

```
ST_GeomFromEWKT('MULTIPOINT(1 0,EMPTY)');
```

srid

Un valore di tipo INTEGER che rappresenta uno Spatial Reference Identifier (SRID). Se è fornito un valore SRID, l'oggetto geometrico restituito presenta tale valore SRID. In caso contrario, il valore SRID dell'oggetto geometrico restituito è impostato a zero (0).

Tipo restituito

GEOMETRY

Se wkt_string o srid sono nulli, allora viene restituito il valore nullo.

Se srid è negativo, allora viene restituito il valore nullo.

Se wkt_string non è valido, allora viene restituito un errore.

Se srid non è valido, allora viene restituito un errore.

Esempi

Il seguente comando SQL costruisce un oggetto geometrico a partire dalla rappresentazione WKT e dal valore SRID.

```
SELECT ST_GeomFromText('POLYGON((0 0,0 1,1 1,1 0,0 0))',4326);
```

```
st_geomfromtext
--------------------------------
0103000020E6100000010000000500000000000000000000000000000000000000000000000000F
```

GeomFromST_ WKB

ST_ GeomFrom WKB costruisce un oggetto geometrico a partire da una rappresentazione binaria esadecimale nota (WKB) di una geometria di input.

ST_ GeomFrom WKB accetta geometrie 3DZ, 3DM e 4D scritte in formato esadecimale WKB.

Sintassi

```
ST_GeomFromWKB(wkb_string)
```

```
ST_GeomFromWKB(wkb_string, srid)
```

Argomenti

wkb_string

Un valore di tipo VARCHAR che è una rappresentazione esadecimale in formato WKB di un oggetto geometrico.

srid

Un valore di tipo INTEGER che rappresenta uno Spatial Reference Identifier (SRID). Se è fornito un valore SRID, l'oggetto geometrico restituito presenta tale valore SRID. In caso contrario, il valore SRID dell'oggetto geometrico restituito è impostato a 0.

Tipo restituito

GEOMETRY

Se wkt_string o srid sono nulli, allora viene restituito il valore nullo.

Se srid è negativo, allora viene restituito il valore nullo.

Se wkb_string non è valido, allora viene restituito un errore.

Se srid non è valido, allora viene restituito un errore.

Esempi

Il seguente comando SQL costruisce un poligono a partire da un valore WKB e restituisce la rappresentazione WKT di un poligono.

```
SELECT
 ST_AsText(ST_GeomFromWKB('010300000001000000050000000000000000000000000000000000000000
```

```
 st_astext
-------------------------------
 POLYGON((0 0,0 1,1 1,1 0,0 0))
```

ST_ GeoSquare

ST_ suddivide GeoSquare ricorsivamente il dominio ([-180, 180], [-90, 90]) in regioni quadrate uguali chiamate geosquadrati fino a una profondità specificata. La suddivisione si basa sulla posizione di un punto fornito. Uno dei valori geosquare contenenti il punto viene suddiviso ad ogni fase fino a raggiungere la profondità massima. La selezione di questo valore geosquare è stabile, cioè il risultato della funzione dipende solo dagli argomenti di input. La funzione restituisce un valore univoco che identifica il valore geosquare finale in cui si trova il punto.

ST_ GeoSquare accetta un PUNTO in cui la coordinata x rappresenta la longitudine e la coordinata y rappresenta la latitudine. La longitudine e la latitudine sono limitate rispettivamente a [-180, 180] e [-90, 90]. L'output di ST_ GeoSquare può essere utilizzato come input per la funzione. ST_ GeomFromGeoSquare

Ci sono 360° attorno all'arco della circonferenza equatoriale della Terra che sono divisi in due emisferi (orientale e occidentale), ciascuno con 180° di linee longitudinali (meridiani) dal meridiano 0°. Per convenzione, le longitudini orientali sono coordinate "+" (positive) quando proiettate su un asse x su un piano cartesiano e le longitudini occidentali sono coordinate "-" (negative) se proiettate su un asse x su un piano cartesiano. Ci sono 90° di linee latitudinali a nord e a sud della circonferenza equatoriale di 0° della Terra, ciascuna parallela alla circonferenza equatoriale di 0° della Terra. Per convenzione, le linee latitudinali settentrionali intersecano l'asse y "+" (positivo) quando vengono proiettate su un piano cartesiano e le linee latitudinali meridionali intersecano l'asse y "-" (negativo) quando vengono proiettate su un piano cartesiano. La griglia sferica formata dall'intersezione di linee longitudinali e latitudinali viene convertita in una griglia proiettata su un piano cartesiano con coordinate x positive e negative standard e coordinate y positive e negative sul piano cartesiano.

Lo scopo di ST_ GeoSquare è etichettare o contrassegnare i punti vicini con valori di codice uguali. I punti che si trovano nella stesso valore geosquare ricevono lo stesso valore di codice. Un valore geosquare viene utilizzato per codificare le coordinate geografiche (latitudine e longitudine) in un numero intero. Una regione più ampia è divisa in griglie per delineare un'area su una mappa con risoluzioni diverse. Un valore geosquare può essere utilizzato per l'indicizzazione spaziale, il binning spaziale, le ricerche di prossimità, la ricerca della posizione e la creazione di identificatori di luoghi univoci. La funzione ST_ GeoHash segue un processo simile di divisione di una regione in griglie, ma ha una codifica diversa.

Sintassi

```
ST_GeoSquare(geom)
```

```
ST_GeoSquare(geom, max_depth)
```

Argomenti

geom

Un valore POINT del tipo di dati GEOMETRY o un'espressione che restituisce un valore di sottotipo POINT. La coordinata x (longitudine) del punto deve essere compresa nell'intervallo: -180 - 180. La coordinata y (latitudine) del punto deve essere compresa nell'intervallo: -90 - 90.

max_depth

Un valore di tipo INTEGER. Il numero massimo di volte in cui il dominio contenente il punto viene suddiviso in modo ricorsivo. Il valore deve essere un numero intero compreso tra 1 e 32. Il valore predefinito è 32. Il numero finale effettivo delle suddivisioni è inferiore o uguale al valore max_depth specificato.

Tipo restituito

BIGINT

La funzione restituisce un valore univoco che identifica il valore geosquare finale in cui si trova il punto di input.

Se l'input geom non è un punto, la funzione restituisce un errore.

Se il punto di input è vuoto, il valore restituito non è un input valido per la funzione ST_ GeomFromGeoSquare. Utilizzate la ST_ IsEmpty funzione per impedire le chiamate a ST_ GeoSquare con un punto vuoto.

Se il punto di input non rientra nell'intervallo, la funzione restituisce un errore.

Se l'input max_depth non rientra nell'intervallo, la funzione restituisce un errore.

Esempi

Il seguente comando SQL restituisce un valore geosquare da un punto di input.

```
SELECT ST_GeoSquare(ST_Point(13.5, 52.5));
```

```
  st_geosquare
-----------------------
 -4410772491521635895
```

Il seguente comando SQL restituisce un valore geosquare da un punto di input con una profondità massima di 10.

```
SELECT ST_GeoSquare(ST_Point(13.5, 52.5), 10);
```

```
 st_geosquare
--------------
 797852
```

ST_ N InteriorRing

ST_ InteriorRing N restituisce una stringa a linee chiuse corrispondente all'anello interno di un poligono di input nella posizione dell'indice. La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_InteriorRingN(geom, index)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

indice

Un valore del tipo di dati INTEGER che rappresenta la posizione di un anello di un indice basato su un uno.

Tipo restituito

`GEOMETRY` di sottotipo `LINESTRING`.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

Se geom o index sono nulli, allora viene restituito il valore nullo.

Se index è fuori intervallo, allora viene restituito il valore nullo.

Se geom non è un poligono, allora viene restituito il valore nullo.

Se geom è un poligono vuoto, allora viene restituito null.

Esempi

Il seguente SQL restituisce il secondo anello del poligono come linestring chiusa.

```
SELECT ST_AsEWKT(ST_InteriorRingN(ST_GeomFromText('POLYGON((7 9,8 7,11 6,15 8,16 6,17
 7,17 10,18 12,17 14,15 15,11 15,10 13,9 12,7 9),(9 9,10 10,11 11,11 10,10 8,9 9),(12
 14,15 14,13 11,12 14))'),2));
```

```
st_asewkt
-----------
 LINESTRING(12 14,15 14,13 11,12 14)
```

ST_Intersects

ST_Intersects restituisce true se le proiezioni 2D delle due geometrie di input hanno almeno un punto in comune.

Sintassi

```
ST_Intersects(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica se il primo poligono interseca il secondo poligono.

```
SELECT ST_Intersects(ST_GeomFromText('POLYGON((0 0,10 0,10 10,0 10,0 0),(2 2,2 5,5 5,5
 2,2 2))'), ST_GeomFromText('MULTIPOINT((4 4),(6 6))'));
```

```
st_intersects
-------------
 true
```

ST_Intersection

ST_intersection restituisce una geometria che rappresenta l'intersezione di due geometrie. Cioè, restituisce la parte delle due geometrie di input condivise tra di esse.

Sintassi

```
ST_Intersection(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

GEOMETRY

Se geom1 e geom2 non condividono alcuno spazio (sono disgiunti), allora viene restituita una geometria vuota.

Se geom1 o geom2 sono vuoti, allora viene restituita una geometria vuota.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Se geom1 o geom2 non è una geometria bidimensionale (2D), allora viene restituito un errore.

Esempi

Il seguente SQL restituisce la geometria non vuota che rappresenta l'intersezione di due geometrie di input.

```
SELECT ST_AsEWKT(ST_Intersection(ST_GeomFromText('polygon((0 0,100 100,0 200,0 0))'),
 ST_GeomFromText('polygon((0 0,10 0,0 10,0 0))')));
```

```
        st_asewkt
-------------------------
 POLYGON((0 0,0 10,5 5,0 0))
```

Il seguente SQL restituisce una geometria vuota quando vengono passate le geometrie di input disgiunte (non intersecanti).

```
SELECT ST_AsEWKT(ST_Intersection(ST_GeomFromText('linestring(0 100,0 0)'),
 ST_GeomFromText('polygon((1 0,10 0,1 10,1 0))')));
```

```
    st_asewkt
------------------
 LINESTRING EMPTY
```

IsPolygonST_ CCW

ST_ IsPolygon CCW restituisce true se la proiezione 2D del poligono o del multipoligono di input è in senso antiorario. Se la geometria di input è un punto, linestring, multipunto o multilinestring, viene restituito true. Per le raccolte di geometrie, ST_ CCW restituisce true se tutte le geometrie della raccolta sono in senso antiorario. IsPolygon

Sintassi

```
ST_IsPolygonCCW(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`BOOLEAN`

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente SQL verifica che il poligono sia in senso antiorario.

```
SELECT ST_IsPolygonCCW(ST_GeomFromText('POLYGON((7 9,8 7,11 6,15 8,16 6,17 7,17 10,18
 12,17 14,15 15,11 15,10 13,9 12,7 9),(9 9,10 10,11 11,11 10,10 8,9 9),(12 14,15 14,13
 11,12 14))'));
```

```
 st_ispolygonccw
----------
 true
```

IsPolygonST_ CW

ST_ IsPolygon CW restituisce true se la proiezione 2D del poligono o del multipoligono di input è in senso orario. Se la geometria di input è un punto, linestring, multipunto o multilinestring, viene restituito true. Per le raccolte di geometrie, IsPolygon ST_ CW restituisce true se tutte le geometrie della raccolta sono in senso orario.

Sintassi

```
ST_IsPolygonCW(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica che il poligono sia in senso orario.

```
SELECT ST_IsPolygonCW(ST_GeomFromText('POLYGON((7 9,8 7,11 6,15 8,16 6,17 7,17 10,18
 12,17 14,15 15,11 15,10 13,9 12,7 9),(9 9,10 10,11 11,11 10,10 8,9 9),(12 14,15 14,13
 11,12 14))'));
```

```
 st_ispolygonccw
----------
 true
```

ST_ IsClosed

ST_ IsClosed restituisce true se la proiezione 2D della geometria di input è chiusa. Un oggetto geometrico chiuso è definito dalle seguenti regole:

- L'oggetto geometrico in input è un punto o un multipunto.
- L'oggetto geometrico è una linestring e il punto iniziale e quello finale della linestring coincidono.
- L'oggetto geometrico in input è una linestring multipla non vuota e tutte le sue linestring sono chiuse.
- L'oggetto geometrico in input è un poligono non vuoto, tutti gli anelli del poligono sono non vuoti e il punto iniziale e quello finale di tutti gli anelli coincidono.
- L'oggetto geometrico in input è un poligono multiplo non vuoto e tutti i suoi poligono sono chiusi.
- L'oggetto geometrico in input è una raccolta di oggetti geometrici non vuota e tutti i suoi componenti sono chiusi.

Sintassi

```
ST_IsClosed(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom è un punto vuoto, allora viene restituito false.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica che il poligono sia chiuso.

```
SELECT ST_IsClosed(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'));
```

```
st_isclosed
-----------
 true
```

ST_ IsCollection

ST_ IsCollection restituisce true se la geometria di input è uno dei seguenti sottotipi:, , o. `GEOMETRYCOLLECTION` `MULTIPOINT` `MULTILINESTRING` `MULTIPOLYGON`

Sintassi

```
ST_IsCollection(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`BOOLEAN`

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica che il poligono sia una raccolta.

```
SELECT ST_IsCollection(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'));
```

```
st_iscollection
-----------
 false
```

ST_ IsEmpty

ST_ IsEmpty restituisce true se la geometria di input è vuota. Una geometria non è vuota se contiene almeno un punto non vuoto.

ST_ IsEmpty restituisce true se la geometria di input ha almeno un punto non vuoto.

Sintassi

```
ST_IsEmpty(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica che il poligono specificato sia vuoto.

```
SELECT ST_IsEmpty(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'));
```

```
st_isempty
-----------
 false
```

ST_ IsRing

ST_ IsRing restituisce true se la stringa di linea di input è un anello. Una linestring è un anello se è chiuso e semplice.

Sintassi

```
ST_IsRing(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. La geometria deve essere un LINESTRING.

Tipo restituito

BOOLEAN

Se geom non è un LINESTRING, allora viene restituito un errore.

Esempi

Il seguente SQL verifica che la linestring specificata sia un anello.

```
SELECT ST_IsRing(ST_GeomFromText('linestring(0 0, 1 1, 1 2, 0 0)'));
```

```
st_isring
-----------
 true
```

ST_ IsSimple

ST_ IsSimple restituisce true se la proiezione 2D della geometria di input è semplice. Per ulteriori informazioni sulla definizione di una geometria semplice, consultare Semplicità geometrica.

Sintassi

```
ST_IsSimple(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente SQL verifica che la linestring specificata sia semplice. In questo esempio, non è semplice perché ha un'auto-intersezione.

```
SELECT ST_IsSimple(ST_GeomFromText('LINESTRING(0 0,10 0,5 5,5 -5)'));
```

```
 st_issimple
-----------
 false
```

ST_ IsValid

ST_ IsValid restituisce true se la proiezione 2D della geometria di input è valida. Per ulteriori informazioni sulla definizione di una geometria valida, consultare Validità geometrica.

Sintassi

```
ST_IsValid(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente SQL verifica che il poligono specificato sia valido. In questo esempio, il poligono non è valido perché l'interno del poligono non è semplicemente connesso.

```
SELECT ST_IsValid(ST_GeomFromText('POLYGON((0 0,10 0,10 10,0 10,0 0),(5 0,10 5,5 10,0
 5,5 0))'));
```

```
 st_isvalid
-----------
 false
```

ST_Length

Per una geometria lineare, ST_Length restituisce la lunghezza cartesiana di una proiezione 2D. Le unità di lunghezza sono le stesse delle unità in cui sono espresse le coordinate della geometria di input. La funzione restituisce zero (0) per punti, multipunti e geometrie areali. Quando l'input è una raccolta di geometrie, la funzione restituisce la somma delle lunghezze delle geometrie nella raccolta.

Per una geografia, ST_Length restituisce la lunghezza geodetica della proiezione 2D di una geografia lineare di input calcolata sullo sferoide determinato dallo SRID. L'unità di lunghezza è espressa in metri. La funzione restituisce zero (0) per punti, multipunti e geografie areali. Quando l'input è una raccolta di geometrie, la funzione restituisce la somma delle lunghezze delle geografie nella raccolta.

Sintassi

```
ST_Length(geo)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

Tipo restituito

DOUBLE PRECISION

Se geo è nullo, allora viene restituito il valore nullo.

Se il valore SRID non viene trovato, allora viene restituito un errore.

Esempi

Il seguente codice SQL restituisce la lunghezza cartesiana di una multilinestring.

```
SELECT ST_Length(ST_GeomFromText('MULTILINESTRING((0 0,10 0,0 10),(10 0,20 0,20
 10))'));
```

```
st_length
--------------------------------
  44.142135623731
```

Il seguente SQL restituisce la lunghezza di una linestring in una geografia.

```
SELECT ST_Length(ST_GeogFromText('SRID=4326;LINESTRING(5 0,6 0,4 0)'));
```

```
 st_length
------------------
 333958.472379804
```

Il seguente SQL restituisce la lunghezza di un punto in una geografia

```
SELECT ST_Length(ST_GeogFromText('SRID=4326;POINT(4 5)'));
```

```
 st_length
-----------
 0
```

ST_ LengthSphere

ST_ LengthSphere restituisce la lunghezza di una geometria lineare in metri. Per le geometrie puntuali, multipunto e areali, ST_ restituisce 0. LengthSphere Per le raccolte di geometrie, ST_ LengthSphere restituisce la lunghezza totale delle geometrie lineari della raccolta in metri.

ST_ LengthSphere interpreta le coordinate di ogni punto della geometria di input come longitudine e latitudine in gradi. Per le geometrie 3DZ, 3DM o 4D, vengono utilizzate solo le prime due coordinate.

Sintassi

```
ST_LengthSphere(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

Lunghezza DOUBLE PRECISION in metri. Il calcolo della lunghezza si basa sul modello sferico della Terra il cui raggio è il raggio medio della Terra del modello ellissoidale 84 della Terra del Sistema Geodetico Mondiale (WGS).

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Nell'esempio seguente SQL calcola la lunghezza di una linestring in metri.

```
SELECT ST_LengthSphere(ST_GeomFromText('LINESTRING(10 10,45 45)'));
```

```
 st_lengthsphere
------------------
 5127736.08292556
```

ST_Length2D

ST_Length2D è un alias per ST_Length. Per ulteriori informazioni, consulta ST_Length.

ST_ LineFromMultiPoint

ST_ LineFromMultiPoint restituisce una stringa di linea da una geometria multipunto di input. L'ordine dei punti è conservato. Il valore dell'identificatore del sistema di riferimento spaziale (SRID)

dell'oggetto geometrico restituito è il valore SRID degli oggetti geometrici in input. La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_LineFromMultiPoint(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere MULTIPOINT.

Tipo restituito

GEOMETRY

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituito il LINESTRING vuoto.

Se geom contiene punti vuoti, questi punti vuoti saranno ignorati.

Se geom non è un MULTIPOINT, viene restituito un errore.

Esempi

Il seguente codice SQL crea una linestring da un multipunto.

```
SELECT ST_AsEWKT(ST_LineFromMultiPoint(ST_GeomFromText('MULTIPOINT(0 0,10 0,10 10,5 5,0
 5)',4326)));
```

```
 st_asewkt
-------------------------------------------
 SRID=4326;LINESTRING(0 0,10 0,10 10,5 5,0 5)
```

ST_ LineInterpolatePoint

ST_ LineInterpolatePoint restituisce un punto lungo una linea a una distanza frazionaria dall'inizio della linea.

Per determinare l'uguaglianza dei punti, ST_ LineInterpolatePoint opera sulla proiezione 2D della geometria di input. Se la geometria di input è vuota, una sua copia viene restituita nella stessa dimensione dell'input. Per le geometrie 3DZ, 3DM e 4D, la coordinata `z` o `m` è la media delle coordinate `z` o `m` del segmento in cui si trova il punto.

Sintassi

```
ST_LineInterpolatePoint(geom, fraction)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Il sottotipo è `LINESTRING`.

fraction

Un valore di tipo di dati `DOUBLE PRECISION` che rappresenta la posizione di un punto lungo la linestring della linea. Il valore è una frazione nell'intervallo 0-1, estremi inclusi.

Tipo restituito

`GEOMETRY` di sottotipo `POINT`.

Se geom o fraction sono null, allora viene restituito null.

Se geom è vuoto, allora viene restituito il punto vuoto.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

Se fraction è fuori intervallo, allora viene restituito un errore.

Se geom non è una linestring, allora viene restituito un errore.

Esempi

Il seguente SQL restituisce un punto a metà strada lungo una linestring.

```
SELECT ST_AsEWKT(ST_LineInterpolatePoint(ST_GeomFromText('LINESTRING(0 0, 5 5, 7 7, 10
 10)'), 0.50));
```

```
st_asewkt
-----------
 POINT(5 5)
```

Il seguente SQL restituisce un punto al 90% sulla strada lungo una linestring.

```
SELECT ST_AsEWKT(ST_LineInterpolatePoint(ST_GeomFromText('LINESTRING(0 0, 5 5, 7 7, 10
 10)'), 0.90));
```

```
st_asewkt
-----------
 POINT(9 9)
```

ST_M

ST_M restituisce la coordinata `m` di un punto in input.

Sintassi

```
ST_M(point)
```

Argomenti

point

Un valore `POINT` di tipo `GEOMETRY`.

Tipo restituito

Valore `DOUBLE PRECISION` della coordinata `m`.

Se point è nullo, allora viene restituito il valore nullo.

Se point è un punto 2D o 3DZ, allora viene restituito null.

Se point è il punto vuoto, allora viene restituito null.

Se point non è un `POINT`, allora viene restituito un errore.

Esempi

Il seguente SQL restituisce la coordinata `m` di un punto in una geometria 3DM.

```
SELECT ST_M(ST_GeomFromEWKT('POINT M (1 2 3)'));
```

```
st_m
-----------
 3
```

Il seguente SQL restituisce la coordinata `m` di un punto in una geometria 4D.

```
SELECT ST_M(ST_GeomFromEWKT('POINT ZM (1 2 3 4)'));
```

```
st_m
-----------
 4
```

ST_ MakeEnvelope

ST_ MakeEnvelope restituisce una geometria come segue:

- Se le coordinate di input specificano un punto, la geometria restituita è un punto.
- Se le coordinate di input specificano una linea, la geometria restituita è una linestring.
- In caso contrario, la geometria restituita è un poligono, dove le coordinate di input specificano gli angoli inferiore sinistro e superiore destro di un box.

Se fornito, il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituita è impostato sul valore SRID di input.

Sintassi

```
ST_MakeEnvelope(xmin, ymin, xmax, ymax)
```

```
ST_MakeEnvelope(xmin, ymin, xmax, ymax, srid)
```

Argomenti

xmin

Un valore di tipo DOUBLE PRECISION. Questo valore è la prima coordinata dell'angolo in basso a sinistra di un box.

ymin

Un valore di tipo DOUBLE PRECISION. Questo valore è la seconda coordinata dell'angolo in basso a sinistra di un box.

xmax

Un valore di tipo DOUBLE PRECISION. Questo valore è la prima coordinata dell'angolo in alto a destra di un box.

ymax

Un valore di tipo DOUBLE PRECISION. Questo valore è la seconda coordinata dell'angolo in alto a destra di un box.

srid

Un valore di tipo di dati INTEGER che rappresenta un identificatore di sistema di riferimento spaziale (SRID). Se il valore SRID non è fornito, allora viene impostato su zero.

Tipo restituito

GEOMETRY di sottotipo POINT, LINESTRING o POLYGON.

Lo SRID della geometria restituita è impostato su srid o zero se srid non è impostato.

Se xmin, ymin, xmax, ymax o srid è null, allora viene restituito il valore null.

Se srid è negativo, allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce un poligono che rappresenta un envelope definito dai quattro valori di coordinate di input.

```
SELECT ST_AsEWKT(ST_MakeEnvelope(2,4,5,7));
```

```
 st_astext
---------------
 POLYGON((2 4,2 7,5 7,5 4,2 4))
```

Il seguente SQL restituisce un poligono che rappresenta un envelope definito dai quattro valori di coordinate di input e un valore SRID.

```
SELECT ST_AsEWKT(ST_MakeEnvelope(2,4,5,7,4326));
```

```
 st_astext
---------------------------------
 SRID=4326;POLYGON((2 4,2 7,5 7,5 4,2 4))
```

ST_ MakeLine

ST_ MakeLine crea una stringa di linea dalle geometrie di input.

La dimensione della geometria restituita è la stessa delle geometrie di input. Entrambe le geometrie di input devono avere la stessa dimensione.

Sintassi

```
ST_MakeLine(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere POINT, LINESTRING o MULTIPOINT.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere POINT, LINESTRING o MULTIPOINT.

Tipo restituito

GEOMETRY di sottotipo LINESTRING.

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 sono il punto vuoto o contengono punti vuoti, allora questi punti vuoti vengono ignorati.

Se geom1 e geom2 sono vuoti, allora viene restituito il `LINESTRING` vuoto.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è il valore SRID degli oggetti geometrici in input.

Se geom1 e geom2 presentano valori SRID diversi, allora viene restituito un errore.

Se geom1 o geom2 non è un `POINT`, `LINESTRING` o `MULTIPOINT`, allora viene restituito un errore.

Se geom1 e geom2 hanno dimensioni diverse, allora viene restituito un errore.

Esempi

Il seguente comando SQL costruisce una linestring a partire da due linestring in input.

```
SELECT ST_MakeLine(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27
 29.31,77.29 29.07)'), ST_GeomFromText('LINESTRING(88.29 39.07,88.42 39.26,88.27
 39.31,88.29 39.07)'));
```

```
st_makeline
-----------
 01020000000800000C3F5285C8F52534052B81E85EB113D407B14AE47E15A5340C3F5285C8F423D40E17A14AE4751
```

ST_ MakePoint

ST_ MakePoint restituisce una geometria puntuale i cui valori di coordinate sono i valori di input.

Sintassi

```
ST_MakePoint(x, y)
```

```
ST_MakePoint(x, y, z)
```

```
ST_MakePoint(x, y, z, m)
```

Argomenti

x

Un valore di tipo DOUBLE PRECISION che rappresenta la prima coordinata.

y

Un valore di tipo DOUBLE PRECISION che rappresenta la seconda coordinata.

z

Un valore di tipo di dati DOUBLE PRECISION che rappresenta la terza coordinata.

m

Un valore di tipo di dati DOUBLE PRECISION che rappresenta la quarta coordinata.

Tipo restituito

GEOMETRY di sottotipo POINT.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è impostato a 0.

Se x, y, z o m è null, allora viene restituito un valore null.

Esempi

Il seguente comando SQL restituisce un tip GEOMETRY di sottotipo POINT con le coordinate specificate.

```
SELECT ST_AsText(ST_MakePoint(1,3));
```

```
st_astext
-----------
 POINT(1 3)
```

Il seguente comando SQL restituisce un tip GEOMETRY di sottotipo POINT con le coordinate specificate.

```
SELECT ST_AsEWKT(ST_MakePoint(1, 2, 3));
```

```
st_asewkt
----------------
 POINT Z (1 2 3)
```

Il seguente comando SQL restituisce un tip `GEOMETRY` di sottotipo `POINT` con le coordinate specificate.

```
SELECT ST_AsEWKT(ST_MakePoint(1, 2, 3, 4));
```

```
st_asewkt
-------------------
 POINT ZM (1 2 3 4)
```

ST_ MakePolygon

ST_ ha due varianti MakePolygon che restituiscono un poligono. Una variante utilizza una singola geometria mentre l'altra ne utilizza due.

- L'input della prima variante è una linestring che definisce l'anello esterno del poligono di output.
- L'input della seconda variante è una linestring e un multilinestring. Entrambi sono vuoti o chiusi.

  Il limite dell'anello esterno del poligono di output è la linestring di ingresso, mentre i confini degli anelli interni del poligono sono le linestring nel multilinestring di input. Se la linestring di input è vuota, viene restituito un poligono vuoto. Le linestring vuote nel multilinestring non vengono prese in considerazione. Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria risultante è il valore SRID comune delle due geometrie di input.

La dimensione della geometria restituita è la stessa delle geometrie di input. L'anello esterno e gli anelli interni devono avere la stessa dimensione.

Sintassi

```
ST_MakePolygon(geom1)
```

```
ST_MakePolygon(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere LINESTRING. Il valore di tipo linestring deve essere chiuso o vuoto.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere MULTILINESTRING.

Tipo restituito

GEOMETRY di sottotipo POLYGON.

L'identificatore del sistema di riferimento spaziale (SRID) della geometria restituita è uguale allo SRID degli input.

Se geom1 o geom2 sono null, allora viene restituito il valore null.

Se geom1 non è una linestring, allora viene restituito un errore.

Se geom2 non è un multilinestring, allora viene restituito un errore.

Se geom1 non è chiuso, allora viene restituito un errore.

Se geom1 è un singolo punto o non è chiuso, allora viene restituito un errore.

Se geom2 contiene almeno una linestring con un singolo punto o non è chiusa, allora viene restituito un errore.

Se geom1 e geom2 presentano valori SRID diversi, allora viene restituito un errore.

Se geom1 e geom2 hanno dimensioni diverse, allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce un poligono a partire da una linestring in input

```
SELECT ST_AsText(ST_MakePolygon(ST_GeomFromText('LINESTRING(77.29 29.07,77.42
 29.26,77.27 29.31,77.29 29.07)')));
```

```
 st_astext
---------------
POLYGON((77.29 29.07,77.42 29.26,77.27 29.31,77.29 29.07))
```

Il seguente SQL crea un poligono da una linestring chiusa e un multilinestring chiuso. La linestring viene utilizzata per l'anello esterno del poligono. I linestring nei multilinestring sono utilizzati per gli anelli interni del poligono.

```
SELECT ST_AsEWKT(ST_MakePolygon(ST_GeomFromText('LINESTRING(0 0,10 0,10 10,0 10,0 0)'),
 ST_GeomFromText('MULTILINESTRING((1 1,1 2,2 1,1 1),(3 3,3 4,4 3,3 3))')));
```

```
 st_astext
---------------------------------
POLYGON((0 0,10 0,10 10,0 10,0 0),(1 1,1 2,2 1,1 1),(3 3,3 4,4 3,3 3))
```

ST_ MemSize

ST_ MemSize restituisce la quantità di spazio di memoria (in byte) utilizzata dalla geometria di input. Questa dimensione dipende dalla rappresentazione interna di Amazon Redshift della geometria e quindi può cambiare se la rappresentazione interna cambia. È possibile utilizzare questa dimensione come indicazione della dimensione relativa degli oggetti geometrici in Amazon Redshift.

Sintassi

```
ST_MemSize(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`INTEGER` rappresentante la dimensione intrinseca di geom.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente codice SQL restituisce la dimensione della memoria di una raccolta di geometrie.

```
SELECT ST_MemSize(ST_GeomFromText('GEOMETRYCOLLECTION(POLYGON((0 0,10 0,0 10,0
 0)),LINESTRING(20 10,20 0,10 0))')::varchar + ' bytes';
```

```
 ?column?
-----------
 172 bytes
```

ST_MMax

ST_MMax restituisce la coordinata `m` massima di una geometria di input.

Sintassi

```
ST_MMax(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

Valore `DOUBLE PRECISION` della coordinata `m` massima.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è una geometria 2D o 3DZ, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce il valore più grande della coordinata `m` di una linestring in una geometria 3DM.

```
SELECT ST_MMax(ST_GeomFromEWKT('LINESTRING M (0 1 2, 3 4 5, 6 7 8)'));
```

```
st_mmax
-----------
  8
```

Il seguente comando SQL restituisce il valore più grande della coordinata `m` di una linestring in una geometria 4D.

```
SELECT ST_MMax(ST_GeomFromEWKT('LINESTRING ZM (0 1 2 3, 4 5 6 7, 8 9 10 11)'));
```

```
st_mmax
-----------
  11
```

ST_MMin

ST_MMin restituisce la coordinata `m` minima di una geometria di input.

Sintassi

```
ST_MMin(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

Valore `DOUBLE PRECISION` della coordinata `m` minima.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è una geometria 2D o 3DZ, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce il valore più piccolo della coordinata `m` di una linestring in una geometria 3DM.

```
SELECT ST_MMin(ST_GeomFromEWKT('LINESTRING M (0 1 2, 3 4 5, 6 7 8)'));
```

```
st_mmin
-----------
  2
```

Il seguente SQL restituisce il valore più piccolo della coordinata `m` di una linestring in una geometria 4D.

```
SELECT ST_MMin(ST_GeomFromEWKT('LINESTRING ZM (0 1 2 3, 4 5 6 7, 8 9 10 11)'));
```

```
st_mmin
-----------
  3
```

ST_Multi

ST_Multi converte una geometria nel multitipo corrispondente. Se la geometria di input è già un multitipo o una raccolta di geometrie, ne viene restituita una copia. Se la geometria di input è un punto, una linestring o un poligono, allora vengono restituiti rispettivamente il multipunto, la multilinestring o il multipoligono.

Sintassi

```
ST_Multi(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

GEOMETRY con sottotipo MULTIPOINT, MULTILINESTRING, MULTIPOLYGON o GEOMETRYCOLLECTION.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è il valore SRID degli oggetti geometrici in input.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente SQL restituisce un multipunto da un multipunto di input.

```
SELECT ST_AsEWKT(ST_Multi(ST_GeomFromText('MULTIPOINT((1 2),(3 4))', 4326)));
```

```
    st_asewkt
-----------------------------------
  SRID=4326;MULTIPOINT((1 2),(3 4))
```

Il seguente SQL restituisce un multipunto da un punto di input.

```
SELECT ST_AsEWKT(ST_Multi(ST_GeomFromText('POINT(1 2)', 4326)));
```

```
    st_asewkt
-----------------------------------
  SRID=4326;MULTIPOINT((1 2))
```

Il seguente SQL restituisce una raccolta di geometrie da una raccolta di geometrie di input.

```
SELECT ST_AsEWKT(ST_Multi(ST_GeomFromText('GEOMETRYCOLLECTION(POINT(1 2),MULTIPOINT((1
 2),(3 4)))', 4326)));
```

```
    st_asewkt
-----------------------------------
  SRID=4326;GEOMETRYCOLLECTION(POINT(1 2),MULTIPOINT((1 2),(3 4)))
```

ST_NDims

ST_NDims restituisce la dimensione delle coordinate di una geometria. ST_NDims non considera la dimensione topologica di una geometria. Al contrario, restituisce un valore costante a seconda della dimensione della geometria.

Sintassi

```
ST_NDims(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`INTEGER` rappresentante la dimensione intrinseca di geom.

Se geom è nullo, allora viene restituito il valore nullo.

I valori restituiti sono i seguenti.

Valore restituito	Dimensione della geometria di input
2	2D
3	3DZ o 3DM
4	4D

Esempi

Il seguente SQL restituisce il numero di dimensioni di un linestring 2D.

```
SELECT ST_NDims(ST_GeomFromText('LINESTRING(0 0,1 1,2 2,0 0)'));
```

```
st_ndims
-------------
 2
```

Il seguente SQL restituisce il numero di dimensioni di un linestring 3DZ.

```
SELECT ST_NDims(ST_GeomFromText('LINESTRING Z(0 0 3,1 1 3,2 2 3,0 0 3)'));
```

```
st_ndims
-------------
 3
```

Il seguente SQL restituisce il numero di dimensioni di un linestring 3DM.

```
SELECT ST_NDims(ST_GeomFromText('LINESTRING M(0 0 4,1 1 4,2 2 4,0 0 4)'));
```

```
st_ndims
-------------
 3
```

Il seguente SQL restituisce il numero di dimensioni di un linestring 4D.

```
SELECT ST_NDims(ST_GeomFromText('LINESTRING ZM(0 0 3 4,1 1 3 4,2 2 3 4,0 0 3 4)'));
```

```
st_ndims
-------------
 4
```

ST_NPoints

ST_NPoints restituisce il numero di punti non vuoti in una geometria o geografia di input.

Sintassi

```
ST_NPoints(geo)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

Tipo restituito

INTEGER

Se geo è un punto vuoto, allora viene restituito 0.

Se geo è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce il numero di punti contenuti in una linestring.

```
SELECT ST_NPoints(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29
 29.07)'));
```

```
st_npoints
-------------
 4
```

Il seguente comando SQL restituisce il numero di punti contenuti in una linestring in una geografia.

```
SELECT ST_NPoints(ST_GeogFromText('LINESTRING(110 40, 2 3, -10 80, -7 9)'));
```

```
st_npoints
-------------
 4
```

ST_NRings

ST_NRings restituisce il numero di anelli contenuti in un oggetto geometrico in input.

Sintassi

```
ST_NRings(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

INTEGER

Se geom è nullo, allora viene restituito il valore nullo.

I valori restituiti sono i seguenti.

Valore restituito	Sottotipo dato di tipo geometry
0	Restituito se geom è di sottotipo POINT, LINESTRING , MULTIPOINT o MULTILINESTRING
Il numero di anelli.	Restituito se geom è di sottotipo POLYGON o MULTIPOLYGON
Il numero di anelli in tutti i componenti	Restituito se geom è di sottotipo GEOMETRYCOLLECTION

Esempi

Il seguente comando SQL restituisce il numero di anelli contenuti in un multipoligono.

```
SELECT ST_NRings(ST_GeomFromText('MULTIPOLYGON(((0 0,10 0,0 10,0 0)),((0 0,-10 0,0
 -10,0 0)))'));
```

```
 st_nrings
-------------
 2
```

ST_ NumGeometries

ST_ NumGeometries restituisce il numero di geometrie in una geometria di input.

Sintassi

```
ST_NumGeometries(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

INTEGER rappresentante il numero di oggetti geometrici in geom.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è una singola geometria vuota, allora viene restituito il valore 0.

Se geom è una singola geometria non vuota, allora viene restituito 1.

Se geom è un sottotipo GEOMETRYCOLLECTION o MULTI, allora viene restituito il numero di oggetti geometrici.

Esempi

Il seguente comando SQL restituisce il numero di oggetti geometrici in una linestring multipla in input.

```
SELECT ST_NumGeometries(ST_GeomFromText('MULTILINESTRING((0 0,1 0,0 5),(3 4,13 26))'));
```

```
st_numgeometries
-------------
```

```
 2
```

ST_ NumInteriorRings

ST_ NumInteriorRings restituisce il numero di anelli in una geometria poligonale di input.

Sintassi

```
ST_NumInteriorRings(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

INTEGER

Se geom è nullo, allora viene restituito il valore nullo.

Se geom non è un poligono, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce il numero di anelli interni contenuti nel poligono in input.

```
SELECT ST_NumInteriorRings(ST_GeomFromText('POLYGON((0 0,100 0,100 100,0 100,0 0),(1
 1,1 5,5 1,1 1),(7 7,7 8,8 7,7 7))'));
```

```
 st_numinteriorrings
-------------
 2
```

ST_ NumPoints

ST_ NumPoints restituisce il numero di punti in una geometria di input.

Sintassi

```
ST_NumPoints(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

INTEGER

Se geom è nullo, allora viene restituito il valore nullo.

Se geom non è un sottotipo LINESTRING, viene restituito null.

Esempi

Il seguente comando SQL restituisce il numero di punti contenuti nella linestring in input.

```
SELECT ST_NumPoints(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27
 29.31,77.29 29.07)'));
```

```
st_numpoints
-------------
4
```

Il seguente SQL restituisce null perché il geom di input non è un sottotipo LINESTRING.

```
SELECT ST_NumPoints(ST_GeomFromText('MULTIPOINT(1 2,3 4)'));
```

```
st_numpoints
-------------
```

ST_Perimeter

Per una geometria areale di input, ST_Perimeter restituisce il perimetro cartesiano (lunghezza del confine) della proiezione 2D. Le unità perimetrali sono le stesse delle unità in cui vengono espresse le coordinate della geometria di input. La funzione restituisce zero (0) per punti, multipunti e geometrie lineari. Quando l'input è una raccolta di geometrie, la funzione restituisce la somma dei perimetri delle geometrie nella raccolta.

Per una geografia di input, ST_Perimeter restituisce il perimetro geodetico (lunghezza del limite) della proiezione 2D di una geografia areale di input calcolata sullo sferoide determinato dallo SRID. L'unità del perimetro è espressa in metri. La funzione restituisce zero (0) per punti, multipunti e geografie lineari. Quando l'input è una raccolta di geometrie, la funzione restituisce la somma dei perimetri delle geografie nella raccolta.

Sintassi

```
ST_Perimeter(geo)
```

Argomenti

geo

Un valore di tipo GEOMETRY o GEOGRAPHY o un'espressione che restituisce un valore di tipo GEOMETRY o GEOGRAPHY.

Tipo restituito

DOUBLE PRECISION

Se geo è nullo, allora viene restituito il valore nullo.

Se il valore SRID non viene trovato, allora viene restituito un errore.

Esempi

Il seguente codice SQL restituisce il perimetro cartesiano di un multipoligono.

```
SELECT ST_Perimeter(ST_GeomFromText('MULTIPOLYGON(((0 0,10 0,0 10,0 0)),((10 0,20 0,20
 10,10 0)))'));
```

```
 st_perimeter
------------------------------
    68.2842712474619
```

Il seguente codice SQL restituisce il perimetro cartesiano di un multipoligono.

```
SELECT ST_Perimeter(ST_GeomFromText('MULTIPOLYGON(((0 0,10 0,0 10,0 0)),((10 0,20 0,20
 10,10 0)))'));
```

```
 st_perimeter
------------------------------
    68.2842712474619
```

Il seguente comando SQL restituisce il perimetro di un poligono in una geografia.

```
SELECT ST_Perimeter(ST_GeogFromText('SRID=4326;POLYGON((0 0,1 0,0 1,0 0))'));
```

```
 st_perimeter
------------------
 378790.428393693
```

Il seguente SQL restituisce il perimetro di una linestring in una geografia.

```
SELECT ST_Perimeter(ST_GeogFromText('SRID=4326;LINESTRING(5 0,10 0)'));
```

```
 st_perimeter
--------------
 0
```

ST_Perimeter2D

ST_Perimeter2D è un alias per ST_Perimeter. Per ulteriori informazioni, consultare ST_Perimeter.

ST_Point

ST_Point restituisce un oggetto geometrico punto a partire dai valori delle coordinate in input.

Sintassi

```
ST_Point(x, y)
```

Argomenti

x

Un valore di tipo DOUBLE PRECISION che rappresenta la prima coordinata.

y

Un valore di tipo DOUBLE PRECISION che rappresenta la seconda coordinata.

Tipo restituito

GEOMETRY di sottotipo POINT.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è impostato a 0.

Se x o y sono nulli, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce un oggetto geometrico punto a partire dalle coordinate in input.

```
SELECT ST_AsText(ST_Point(5.0, 7.0));
```

```
st_astext
-------------
POINT(5 7)
```

ST_PointN

ST_PointN restituisce un punto in una linestring come specificato da un valore di indice. I valori di indice negativi vengono conteggiati all'indietro dalla fine della linestring, in modo che -1 sia l'ultimo punto.

La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_PointN(geom, index)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere LINESTRING.

indice

Valore del tipo di dati INTEGER che rappresenta l'indice di un punto in una linestring.

Tipo restituito

GEOMETRY di sottotipo POINT.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è impostato a 0.

Se geom o index sono nulli, allora viene restituito il valore nullo.

Se index è fuori intervallo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom non è un LINESTRING, allora viene restituito il valore nullo.

Esempi

Il seguente codice SQL restituisce una rappresentazione estesa di testo noto (EWKT) di una LINESTRING a sei punti a un oggetto GEOMETRY e restituisce il punto all'indice 5 della linestring.

```
SELECT ST_AsEWKT(ST_PointN(ST_GeomFromText('LINESTRING(0 0,10 0,10 10,5 5,0 5,0
 0)',4326), 5));
```

```
st_asewkt
-------------
```

```
 SRID=4326;POINT(0 5)
```

ST_Points

ST_Points restituisce una geometria multipunto contenente tutti i punti non vuoti nella geometria di input. ST_Points non rimuove i punti duplicati nell'input, inclusi i punti iniziale e finale delle geometrie ad anello.

Sintassi

```
ST_Points(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

GEOMETRY di sottotipo MULTIPOINT.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituita è uguale a quello di geom.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituito il multipunto vuoto.

Esempi

Gli esempi SQL seguenti costruiscono una geometria multipunto dalla geometria di input. Il risultato è una geometria multipunto contenente tutti i punti non vuoti nella geometria di input.

```
SELECT ST_AsEWKT(ST_Points(ST_SetSRID(ST_GeomFromText('LINESTRING(1 0,2 0,3 0)'),
 4326)));
```

```
st_asewkt
-------------
```

```
SRID=4326;MULTIPOINT((1 0),(2 0),(3 0))
```

```
SELECT ST_AsEWKT(ST_Points(ST_SetSRID(ST_GeomFromText('MULTIPOLYGON(((0 0,1 0,0 1,0
 0)))'), 4326)));
```

```
st_asewkt
-------------
SRID=4326;MULTIPOINT((0 0),(1 0),(0 1),(0 0))
```

ST_Polygon

ST_Polygon restituisce un oggetto geometrico poligono il cui anello esterno è la linea spezzata in input con il valore che è stato fornito come identificatore del sistema di riferimento spaziale (SRID).

La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_Polygon(linestring, srid)
```

Argomenti

linestring

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Il sottotipo deve essere `LINESTRING` che rappresenta una linestring. Il valore di tipo linestring deve essere chiuso.

srid

Un valore di tipo `INTEGER` che rappresenta un SRID.

Tipo restituito

`GEOMETRY` di sottotipo `POLYGON`.

Il valore SRID dell'oggetto geometrico restituito è impostato a srid.

Se linestring o srid sono nulli, allora viene restituito il valore nullo.

Se linestring non è una linestring, allora viene restituito un errore.

Se linestring non è una linestring, allora viene restituito un errore.

Se srid è negativo, allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce un poligono con un valore SRID.

```
SELECT ST_AsEWKT(ST_Polygon(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27
 29.31,77.29 29.07)'),4356));
```

```
st_asewkt
-------------
 SRID=4356;POLYGON((77.29 29.07,77.42 29.26,77.27 29.31,77.29 29.07))
```

ST_ RemovePoint

ST_ RemovePoint restituisce una geometria di stringhe di linee a cui è stato rimosso il punto della geometria di input in una posizione di indice.

L'indice è a base zero. L'identificatore del sistema di riferimento spaziale (SRID) del risultato è lo stesso della geometria di input. La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_RemovePoint(geom, index)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo sottotipo deve essere `LINESTRING`.

indice

Valore del tipo di dati `INTEGER` che rappresenta la posizione di un indice a base zero.

Tipo restituito

GEOMETRY

Se geom o index sono nulli, allora viene restituito il valore nullo.

Se geom non è sottotipo LINESTRING, viene restituito un errore.

Se index è fuori intervallo, viene restituito un errore. I valori validi per la posizione dell'indice sono compresi tra 0 e ST_NumPoints(geom) meno 1.

Esempi

Il seguente codice SQL rimuove l'ultimo punto di una linestring.

```
WITH tmp(g) AS (SELECT ST_GeomFromText('LINESTRING(0 0,10 0,10 10,5 5,0 5)',4326))
SELECT ST_AsEWKT(ST_RemovePoint(g, ST_NumPoints(g) - 1)) FROM tmp;
```

```
   st_asewkt
----------------------------------------
 SRID=4326;LINESTRING(0 0,10 0,10 10,5 5)
```

ST_Reverse

ST_Reverse inverte l'ordine dei vertici per le geometrie lineari e areali. Per le geometrie a punto o multipunto, viene restituita una copia della geometria originale. Per le raccolte di geometrie, ST_Reverse inverte l'ordine dei vertici per ciascuna geometria nella raccolta.

La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_Reverse(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

GEOMETRY

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è il valore SRID degli oggetti geometrici in input.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL inverte l'ordine dei punti in una linestring.

```
SELECT ST_AsEWKT(ST_Reverse(ST_GeomFromText('LINESTRING(1 0,2 0,3 0,4 0)', 4326)));
```

```
    st_asewkt
-----------------------------------
  SRID=4326;LINESTRING(4 0,3 0,2 0,1 0)
```

ST_ SetPoint

ST_ SetPoint restituisce una stringa di linee con coordinate aggiornate rispetto alla posizione della stringa di linea di input specificata dall'indice. Le nuove coordinate sono le coordinate del punto di input.

La dimensione della geometria restituita è la stessa del valore geom1. Se geom1 e geom2 hanno dimensioni diverse, geom2 è proiettato sulla dimensione di geom1.

Sintassi

```
ST_SetPoint(geom1, index, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere LINESTRING.

indice

Un valore del tipo di dati `INTEGER` che rappresenta la posizione di un indice. Un `0` si riferisce al primo punto del linestring da sinistra, `1` si riferisce al secondo punto e così via. L'indice può essere un valore negativo. Un `-1` si riferisce al primo punto del linestring da sinistra, `-2` si riferisce al secondo punto da destra e così via.

geom2

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`. Questo sottotipo deve essere `POINT`.

Tipo restituito

`GEOMETRY`

Se geom2 è il punto vuoto, allora viene restituito geom1.

Se geom1, geom2 o index sono nulli, allora viene restituito il valore nullo.

Se geom1 non è una linestring, allora viene restituito un errore.

Se index non è compreso in un intervallo di indice valido, allora viene restituito un errore.

Se geom2 non è un punto, allora viene restituito un errore.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Esempi

Il seguente SQL restituisce un nuovo linestring in cui è stato impostato il secondo punto del linestring di input con il punto specificato.

```
SELECT ST_AsText(ST_SetPoint(ST_GeomFromText('LINESTRING(1 2, 3 2, 5 2, 1 2)'), 2,
 ST_GeomFromText('POINT(7 9)')));
```

```
st_astext
-------------
```

```
 LINESTRING(1 2,3 2,7 9,1 2)
```

Il seguente SQL di esempio restituisce un nuovo linestring in cui è stato impostato il terzo punto dalla destra (l'indice è negativo) del linestring di input con il punto specificato.

```
SELECT ST_AsText(ST_SetPoint(ST_GeomFromText('LINESTRING(1 2, 3 2, 5 2, 1 2)'), -3,
 ST_GeomFromText('POINT(7 9)')));
```

```
st_astext
-------------
 LINESTRING(1 2,7 9,5 2,1 2)
```

ST_SetSRID

ST_SetSRID restituisce un oggetto geometrico uguale all'oggetto geometrico in input, salvo essere aggiornato col valore in input per l'identificatore del sistema di riferimento spaziale (SRID).

Sintassi

```
ST_SetSRID(geom, srid)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

srid

Un valore di tipo `INTEGER` che rappresenta un SRID.

Tipo restituito

`GEOMETRY`

Il valore SRID dell'oggetto geometrico restituito è impostato a srid.

Se geom o srid sono nulli, allora viene restituito il valore nullo.

Se srid è negativo, allora viene restituito un errore.

Esempi

Il seguente comando SQL imposta il valore SRID di una linestring.

```
SELECT ST_AsEWKT(ST_SetSRID(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27
 29.31,77.29 29.07)'),50));
```

```
st_asewkt
-------------
 SRID=50;LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29 29.07)
```

ST_Simplify

ST_Simplify restituisce una copia semplificata della geometria di input utilizzando l'algoritmo Ramer-Douglas-Peucker con la tolleranza specificata. La topologia della geometria di input potrebbe non essere mantenuta. Per ulteriori informazioni sull'algoritmo, consultare Algoritmo Ramer-Douglas-Peucker in Wikipedia.

Quando ST_Simplify calcola le distanze per semplificare una geometria, ST_Simplify opera sulla proiezione 2D della geometria di input.

Sintassi

```
ST_Simplify(geom, tolerance)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

tolerance

Un valore di tipo di dati `DOUBLE PRECISION` che rappresenta il livello di tolleranza dell'algoritmo Ramer-Douglas-Peucker. Se tolerance è un numero negativo, allora viene utilizzato il valore null.

Tipo restituito

`GEOMETRY`.

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) della geometria restituito è il valore SRID della geometria di input.

La dimensione della geometria restituita è la stessa della geometria di input.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente SQL semplifica la linestring di input utilizzando una tolleranza di distanza euclidea di 1 con l'algoritmo Ramer-Douglas-Peucker. Le unità della distanza sono uguali a quelle delle coordinate della geometria.

```
SELECT ST_AsEWKT(ST_Simplify(ST_GeomFromText('LINESTRING(0 0,1 2,1 1,2 2,2 1)'), 1));
```

```
 st_asewkt
-----------
LINESTRING(0 0,1 2,2 1)
```

ST_SRID

ST_SRID restituisce l'identificatore del sistema di riferimento spaziale (SRID) di un oggetto geometrico in input. Per ulteriori informazioni su un valore SRID, consulta Query su dati spaziali in Amazon Redshift.

Sintassi

```
ST_SRID(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

`INTEGER` rappresentante il valore del SRID di geom.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce un valore SRID di una linestring impostata su SRID 4326.

```
SELECT ST_SRID(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29
 29.07)',4326));
```

```
st_srid
-------------
 4326
```

Il seguente comando SQL restituisce un valore SRID di una linestring non impostata durante la costruzione. Il risultato è 0 per il valore SRID.

```
SELECT ST_SRID(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29
 29.07)'));
```

```
st_srid
-------------
 0
```

ST_ StartPoint

ST_ StartPoint restituisce il primo punto di una stringa di linee di input. Il valore dell'identificatore del sistema di riferimento spaziale (SRID) del risultato è lo stesso della geometria di input. La dimensione della geometria restituita è la stessa della geometria di input.

Sintassi

```
ST_StartPoint(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo sottotipo deve essere LINESTRING.

Tipo restituito

GEOMETRY

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom non è un LINESTRING, viene restituito il valore nullo.

Esempi

Il seguente codice SQL restituisce una rappresentazione estesa di testo noto (EWKT) di una LINESTRING a quattro punti a un oggetto GEOMETRY e restituisce il punto iniziale della linestring.

```
SELECT ST_AsEWKT(ST_StartPoint(ST_GeomFromText('LINESTRING(0 0,10 0,10 10,5 5,0
 5)',4326)));
```

```
st_asewkt
-------------
 SRID=4326;POINT(0 0)
```

ST_Touches

ST_Touches restituisce il valore true se le proiezioni 2D delle due geometrie di input si toccano. Le due geometrie si toccano se non sono vuote, si intersecano e non hanno punti interni in comune.

Sintassi

```
ST_Touches(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

`BOOLEAN`

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente codice SQL controlla se un poligono tocca una linestring.

```
SELECT ST_Touches(ST_GeomFromText('POLYGON((0 0,10 0,0 10,0 0))'),
 ST_GeomFromText('LINESTRING(20 10,20 0,10 0)'));
```

```
 st_touches
-------------
 t
```

ST_Transform

ST_Transform restituisce una nuova geometria con coordinate trasformate in un sistema di riferimento spaziale definito dall'identificatore di sistema di riferimento spaziale di input (SRID).

Sintassi

```
ST_Transform(geom, srid)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

srid

Un valore di tipo `INTEGER` che rappresenta un SRID.

Tipo restituito

`GEOMETRY`.

Il valore SRID dell'oggetto geometrico restituito è impostato a srid.

Se geom o srid sono nulli, allora viene restituito il valore nullo.

Se il valore SRID associato all'input geom non esiste, quindi viene restituito un errore.

Se srid non esiste, allora viene restituito un errore.

Esempi

Il seguente SQL trasforma lo SRID di una raccolta di geometrie vuote.

```
SELECT ST_AsEWKT(ST_Transform(ST_GeomFromText('GEOMETRYCOLLECTION EMPTY', 3857),
 4326));
```

```
            st_asewkt
-----------------------------------
 SRID=4326;GEOMETRYCOLLECTION EMPTY
```

Il seguente comando SQL trasforma il valore SRID di una linestring.

```
SELECT ST_AsEWKT(ST_Transform(ST_GeomFromText('LINESTRING(110 40, 2 3, -10 80, -7 9,
 -22 -33)', 4326), 26918));
```

```
    st_asewkt
-----------------------------------------------------------------------------------------------
 SRID=26918;LINESTRING(73106.6977300955 15556182.9688576,14347201.5059964
 1545178.32934967,1515090.41262989 9522193.25115316,10491250.83295
 2575457.28410878,5672303.72135968 -5233682.61176205)
```

Il seguente comando SQL trasforma lo SRID di un poligono.

```
SELECT ST_AsEWKT(ST_Transform(ST_GeomFromText('POLYGON Z ((-10 10 -7, -65 10 -6, -10 64
 -5, -10 10 -7), (-11 11 5, -11 12 6, -12 11 7, -11 11 5))', 6989), 6317));
```

```
                                        st_asewkt
-------------------------------------------------------------------------------------------------
 SRID=6317;POLYGON Z ((6186430.2771091 -1090834.57212608
 1100247.33216237,2654831.67853801 -5693304.90741276 1100247.50581055,2760987.41750022
 -486836.575101877 5709710.44137268,6186430.2771091 -1090834.57212608
 1100247.33216237),(6146675.25029258 -1194792.63532103 1209007.1115113,6125027.87562215
 -1190584.81194058 1317403.77865723,6124888.99555252 -1301885.3455052
 1209007.49312929,6146675.25029258 -1194792.63532103 1209007.1115113))
```

ST_Union

ST_Union restituisce una geometria che rappresenta l'unione di due geometrie. Vale a dire, unisce le geometrie di input per produrre una geometria risultante senza sovrapposizioni.

Sintassi

```
ST_Union(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

GEOMETRY

Il valore dell'identificatore del sistema di riferimento spaziale (SRID) dell'oggetto geometrico restituito è il valore SRID degli oggetti geometrici in input.

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 o geom2 sono vuoti, allora viene restituita una geometria vuota.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una raccolta di geometrie, linestring o multilinestring allora viene restituito un errore.

Se geom1 o geom2 non è una geometria bidimensionale (2D), allora viene restituito un errore.

Esempi

La seguente istruzione SQL restituisce la geometria non vuota che rappresenta l'intersezione di due geometrie di input.

```
SELECT ST_AsEWKT(ST_Union(ST_GeomFromText('POLYGON((0 0,100 100,0 200,0 0))'),
 ST_GeomFromText('POLYGON((0 0,10 0,0 10,0 0))')));
```

```
        st_asewkt
-------------------------
 POLYGON((0 0,0 200,100 100,5 5,10 0,0 0))
```

ST_Within

ST_Within restituisce true se la proiezione 2D della prima geometria di input è all'interno della proiezione 2D della seconda geometria di input.

Ad esempio, l'oggetto geometrico A è contenuto all'interno dell'oggetto geometrico B se ogni punto di A è un punto di B e le loro aree interne presentano un'intersezione non vuota.

ST_Within(A, B) è equivalente a ST_Contains(B, A).

Sintassi

```
ST_Within(geom1, geom2)
```

Argomenti

geom1

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY. Questo valore è confrontato con geom2 per determinare se è contenuto all'interno di geom2.

geom2

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom1 o geom2 sono nulli, allora viene restituito il valore nullo.

Se geom1 e geom2 non presentano lo stesso valore di identificatore del sistema di riferimento spaziale (SRID), allora viene restituito il valore nullo.

Se geom1 o geom2 sono una collezione di geometrie, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL verifica se il primo poligono si trova all'interno del secondo poligono.

```
SELECT ST_Within(ST_GeomFromText('POLYGON((0 2,1 1,0 -1,0 2))'),
 ST_GeomFromText('POLYGON((-1 3,2 1,0 -3,-1 3))'));
```

```
st_within
-----------
 true
```

ST_X

ST_X restituisce la prima coordinata di un punto in input.

Sintassi

```
ST_X(point)
```

Argomenti

point

Un valore POINT di tipo GEOMETRY.

Tipo restituito

Valore `DOUBLE PRECISION` della prima coordinata.

Se point è nullo, allora viene restituito il valore nullo.

Se point è il punto vuoto, allora viene restituito null.

Se point non è un `POINT`, allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce la prima coordinata di un punto.

```
SELECT ST_X(ST_Point(1,2));
```

```
st_x
-----------
 1.0
```

ST_XMax

ST_XMax restituisce il massimo valore della prima coordinata di un oggetto geometrico in input.

Sintassi

```
ST_XMax(geom)
```

Argomenti

geom

Un valore di tipo `GEOMETRY` o un'espressione che restituisce un valore di tipo `GEOMETRY`.

Tipo restituito

Valore `DOUBLE PRECISION` del valore massimo della prima coordinata.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce il valore maggiore della prima coordinata di una linestring.

```
SELECT ST_XMax(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29
 29.07)'));
```

```
st_xmax
-----------
 77.42
```

ST_XMin

ST_XMin restituisce il minimo valore della prima coordinata di un oggetto geometrico in input.

Sintassi

```
ST_XMin(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

Valore DOUBLE PRECISION del valore minimo della prima coordinata.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce il valore minore della prima coordinata di una linestring.

```
SELECT ST_XMin(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29
 29.07)'));
```

```
st_xmin
-----------
 77.27
```

ST_Y

ST_X restituisce la seconda coordinata di un punto in input.

Sintassi

```
ST_Y(point)
```

Argomenti

point

Un valore POINT di tipo GEOMETRY.

Tipo restituito

Valore DOUBLE PRECISION della seconda coordinata.

Se point è nullo, allora viene restituito il valore nullo.

Se point è il punto vuoto, allora viene restituito null.

Se point non è un POINT, allora viene restituito un errore.

Esempi

Il seguente comando SQL restituisce la seconda coordinata di un punto.

```
SELECT ST_Y(ST_Point(1,2));
```

```
st_y
-----------
 2.0
```

ST_YMax

ST_XMax restituisce il massimo valore della seconda coordinata di un oggetto geometrico in input.

Sintassi

```
ST_YMax(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

Valore DOUBLE PRECISION del valore massimo della seconda coordinata.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce il valore maggiore della seconda coordinata di una linestring.

```
SELECT ST_YMax(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29
 29.07)'));
```

```
st_ymax
-----------
 29.31
```

ST_YMin

ST_YMin restituisce il massimo valore della seconda coordinata di un oggetto geometrico in input.

Sintassi

```
ST_YMin(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

Valore DOUBLE PRECISION del valore minimo della seconda coordinata.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente comando SQL restituisce il valore minore della seconda coordinata di una linestring.

```
SELECT ST_YMin(ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29
 29.07)'));
```

```
st_ymin
-----------
 29.07
```

ST_Z

ST_Z restituisce la coordinata z di un punto in input.

Sintassi

```
ST_Z(point)
```

Argomenti

point

Un valore POINT di tipo GEOMETRY.

Tipo restituito

Valore `DOUBLE PRECISION` della coordinata `m`.

Se point è nullo, allora viene restituito il valore nullo.

Se point è un punto 2D o 3DM, allora viene restituito null.

Se point è il punto vuoto, allora viene restituito null.

Se point non è un `POINT`, allora viene restituito un errore.

Esempi

Il seguente SQL restituisce la coordinata z di un punto in una geometria 3DZ.

```
SELECT ST_Z(ST_GeomFromEWKT('POINT Z (1 2 3)'));
```

```
st_z
-----------
 3
```

Il seguente SQL restituisce la coordinata z di un punto in una geometria 4D.

```
SELECT ST_Z(ST_GeomFromEWKT('POINT ZM (1 2 3 4)'));
```

```
st_z
-----------
 3
```

ST_ZMax

ST_ZMax restituisce la coordinata z massima di una geometria di input.

Sintassi

```
ST_ZMax(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

Valore DOUBLE PRECISION della coordinata z massima.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è una geometria 2D o 3DM, allora viene restituito null.

Esempi

Il seguente comando SQL restituisce il valore più piccolo della coordinata z di una linestring in una geometria 3DZ.

```
SELECT ST_ZMax(ST_GeomFromEWKT('LINESTRING Z (0 1 2, 3 4 5, 6 7 8)'));
```

```
st_zmax
-----------
  8
```

Il seguente comando SQL restituisce il valore più grande della coordinata z di una linestring in una geometria 4D.

```
SELECT ST_ZMax(ST_GeomFromEWKT('LINESTRING ZM (0 1 2 3, 4 5 6 7, 8 9 10 11)'));
```

```
st_zmax
-----------
  10
```

ST_ZMin

ST_ZMin restituisce la coordinata z minima di una geometria di input.

Sintassi

```
ST_ZMin(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

Valore DOUBLE PRECISION della coordinata z minima.

Se geom è vuoto, allora viene restituito il valore nullo.

Se geom è nullo, allora viene restituito il valore nullo.

Se geom è una geometria 2D o 3DM, allora viene restituito null.

Esempi

Il seguente SQL restituisce il valore più piccolo della coordinata z di una linestring in una geometria 3DZ.

```
SELECT ST_ZMin(ST_GeomFromEWKT('LINESTRING Z (0 1 2, 3 4 5, 6 7 8)'));
```

```
st_zmin
-----------
  2
```

Il seguente SQL restituisce il valore più piccolo della coordinata z di una linestring in una geometria 4D.

```
SELECT ST_ZMin(ST_GeomFromEWKT('LINESTRING ZM (0 1 2 3, 4 5 6 7, 8 9 10 11)'));
```

```
st_zmin
```

```
-----------
  2
```

SupportsBBox

SupportsBBox restituisce true se la geometria di input supporta la codifica con un bounding box precalcolato. Per ulteriori informazioni sul supporto per i bounding box, consultare Riquadro di delimitazione.

Sintassi

```
SupportsBBox(geom)
```

Argomenti

geom

Un valore di tipo GEOMETRY o un'espressione che restituisce un valore di tipo GEOMETRY.

Tipo restituito

BOOLEAN

Se geom è nullo, allora viene restituito il valore nullo.

Esempi

Il seguente SQL restituisce true se la geometria punto di input supporta la codifica con un bounding box.

```
SELECT SupportsBBox(AddBBox(ST_GeomFromText('POLYGON((0 0,1 0,0 1,0 0))')));
```

```
supportsbbox
--------------
t
```

Il seguente SQL restituisce false se la geometria punto di input non supporta la codifica con un bounding box.

```
SELECT SupportsBBox(DropBBox(ST_GeomFromText('POLYGON((0 0,1 0,0 1,0 0))')));
```

```
supportsbbox
--------------
f
```

Funzioni stringa

Argomenti

- Funzione OCTET_LENGTH
- Funzione POSITION
- Funzione QUOTE_IDENT
- Funzione QUOTE_LITERAL
- Funzione REGEXP_COUNT
- Funzione REGEXP_INSTR
- Funzione REGEXP_REPLACE
- Funzione REGEXP_SUBSTR
- Funzione REPEAT
- Funzione REPLACE
- Funzione REPLICATE
- Funzione REVERSE
- Funzione RTRIM
- Funzione SOUNDEX
- Funzione SPLIT_PART
- Funzione STRPOS
- Funzione STRTOL
- Funzione SUBSTRING
- Funzione TEXTLEN
- Funzione TRANSLATE
- Funzione TRIM
- Funzione UPPER

Le funzioni di stringa elaborano e manipolano stringhe di caratteri o espressioni che valutano le stringhe di caratteri. Quando l'argomento stringa in queste funzioni è un valore letterale, deve essere racchiuso tra virgolette singole. I tipi di dati supportati includono CHAR e VARCHAR.

La seguente sezione fornisce i nomi della funzione, la sintassi e le descrizioni per le funzioni supportate. Tutti gli offset in stringhe sono basati su uno.

Funzioni solo sul nodo principale obsolete

Le seguenti funzioni di stringa sono deprecate perché vengono eseguite solo sul nodo principale. Per ulteriori informazioni, consultare Nodo principale: solo funzioni

- GET_BYTE
- SET_BIT
- SET_BYTE
- TO_ASCII

|| (Concatenamento) Operatore

Concatena due espressioni su entrambi i lati del simbolo `||` e restituisce l'espressione concatenata.

Simile a Funzione CONCAT.

> **Note**
>
> Se una o entrambe le espressioni sono nulle, il risultato della concatenazione è `NULL`.

Sintassi

```
expression1 || expression2
```

Argomenti

expression1

Una stringa `CHAR`, una stringa `VARCHAR`, un'espressione binaria o un'espressione che restituisce uno di questi tipi.

expression2

Una stringa `CHAR`, una stringa `VARCHAR`, un'espressione binaria o un'espressione che restituisce uno di questi tipi.

Tipo restituito

Il tipo di dati della stringa è lo stesso tipo degli argomenti di input. Ad esempio, concatenando due stringhe di tipo `VARCHAR` restituisce una stringa di tipo `VARCHAR`.

Esempi

Gli esempi seguenti utilizzano la tabella USERS e VENUE dal database di esempio di TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per concatenare i campi FIRSTNAME e LASTNAME dalla tabella USERS nel database di esempio, usa l'esempio seguente.

```
SELECT (firstname || ' ' || lastname) as fullname
FROM users
ORDER BY 1
LIMIT 10;

+-----------------+
|    fullname     |
+-----------------+
| Aaron Banks     |
| Aaron Booth     |
| Aaron Browning  |
| Aaron Burnett   |
| Aaron Casey     |
| Aaron Cash      |
| Aaron Castro    |
| Aaron Dickerson |
| Aaron Dixon     |
| Aaron Dotson    |
+-----------------+
```

Per concatenare le colonne che potrebbero contenere valori null, utilizzare l'espressione Funzioni NVL e COALESCE. Il seguente esempio utilizza NVL per restituire uno `0` ogni volta che si incontra NULL.

```
SELECT (venuename || ' seats ' || NVL(venueseats, 0)) as seating
FROM venue
WHERE venuestate = 'NV' or venuestate = 'NC'
ORDER BY 1
LIMIT 10;

+-------------------------------------+
|               seating               |
+-------------------------------------+
| Ballys Hotel seats 0                |
```

```
| Bank of America Stadium seats 73298 |
| Bellagio Hotel seats 0              |
| Caesars Palace seats 0              |
| Harrahs Hotel seats 0               |
| Hilton Hotel seats 0                |
| Luxor Hotel seats 0                 |
| Mandalay Bay Hotel seats 0          |
| Mirage Hotel seats 0                |
| New York New York seats 0           |
+-------------------------------------+
```

Funzione ASCII

La funzione ASCII restituisce il codice ASCII, o il punto di codice Unicode, del primo carattere della stringa specificata. Se la stringa è vuota la funzione restituisce `0`. Restituisce `NULL` se la stringa è null.

Sintassi

```
ASCII('string')
```

Argomento

stringa

Una stringa `CHAR` o una stringa `VARCHAR`.

Tipo restituito

INTEGER

Esempi

Per restituire `NULL`, utilizza l'esempio seguente. La funzione NULLIF restituisce `NULL` se i due argomenti sono uguali, pertanto l'argomento di input per la funzione ASCII è `NULL`. Per ulteriori informazioni, consulta Funzione NULLIF.

```
SELECT ASCII(NULLIF('',''));

+-------+
| ascii |
+-------+
```

```
|  NULL |
+-------+
```

Per restituire il codice ASCII 0, utilizza l'esempio seguente.

```
SELECT ASCII('');

+-------+
| ascii |
+-------+
|     0 |
+-------+
```

Per restituire il codice ASCII 97 per la prima lettera della parola amazon, utilizza l'esempio seguente.

```
SELECT ASCII('amazon');

+-------+
| ascii |
+-------+
|    97 |
+-------+
```

Per restituire il codice ASCII 65 per la prima lettera della parola Amazon, utilizza l'esempio seguente.

```
SELECT ASCII('Amazon');

+-------+
| ascii |
+-------+
|    65 |
+-------+
```

Funzione BPCHARCMP

Confronta il valore di due stringhe e restituisce un integer. Se le stringhe sono identiche, la funzione restituisce `0`. Se la prima stringa è alfabeticamente posteriore, la funzione restituisce `1`. Se la seconda stringa è maggiore, la funzione restituisce `-1`.

Per i caratteri multibyte, il confronto si basa sulla codifica dei byte.

Sinonimo di Funzione BTTEXT_PATTERN_CMP.

Sintassi

```
BPCHARCMP(string1, string2)
```

Argomenti

string1

Una stringa CHAR o una stringa VARCHAR.

string2

Una stringa CHAR o una stringa VARCHAR.

Tipo restituito

INTEGER

Esempi

Gli esempi seguenti utilizzano la tabella USERS dal database di esempio di TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per determinare se il nome di un utente è alfabeticamente maggiore rispetto al cognome dell'utente per le prime dieci voci nella tabella USERS, utilizza l'esempio seguente. È possibile vedere che per le voci in cui la stringa per FIRSTNAME è successiva in ordine alfabetico rispetto a LASTNAME, la funzione restituisce 1. Se LASTNAME in ordine alfabetico viene dopo FIRSTNAME, la funzione restituisce -1.

```
SELECT userid, firstname, lastname, BPCHARCMP(firstname, lastname)
FROM users
ORDER BY 1, 2, 3, 4
LIMIT 10;

+--------+-----------+-----------+-----------+
| userid | firstname | lastname  | bpcharcmp |
+--------+-----------+-----------+-----------+
|      1 | Rafael    | Taylor    |        -1 |
|      2 | Vladimir  | Humphrey  |         1 |
|      3 | Lars      | Ratliff   |        -1 |
|      4 | Barry     | Roy       |        -1 |
|      5 | Reagan    | Hodge     |         1 |
|      6 | Victor    | Hernandez |         1 |
```

```
|      7 | Tamekah   | Juarez    |         1 |
|      8 | Colton    | Roy       |        -1 |
|      9 | Mufutau   | Watkins   |        -1 |
|     10 | Naida     | Calderon  |         1 |
+--------+-----------+-----------+-----------+
```

Per restituire tutte le voci nella tabella USERS in cui la funzione restituisce `0`, utilizza l'esempio seguente. La funzione restituisce `0` quando FIRSTNAME è identico a LASTNAME.

```
SELECT userid, firstname, lastname,
BPCHARCMP(firstname, lastname)
FROM users
WHERE BPCHARCMP(firstname, lastname)=0
ORDER BY 1, 2, 3, 4;

+--------+-----------+----------+-----------+
| userid | firstname | lastname | bpcharcmp |
+--------+-----------+----------+-----------+
|     62 | Chase     | Chase    |         0 |
|   4008 | Whitney   | Whitney  |         0 |
|  12516 | Graham    | Graham   |         0 |
|  13570 | Harper    | Harper   |         0 |
|  16712 | Cooper    | Cooper   |         0 |
|  18359 | Chase     | Chase    |         0 |
|  27530 | Bradley   | Bradley  |         0 |
|  31204 | Harding   | Harding  |         0 |
+--------+-----------+----------+-----------+
```

Funzione BTRIM

La funzione BTRIM riduce una stringa rimuovendo spazi vuoti iniziali e finali o rimuovendo i caratteri iniziali e finali che corrispondono a una stringa specificata facoltativa.

Sintassi

```
BTRIM(string [, trim_chars ] )
```

Argomenti

stringa

La stringa VARCHAR di input da ridurre.

trim_chars

La stringa VARCHAR contenente i caratteri da abbinare.

Tipo restituito

La funzione BTRIM restituisce una stringa VARCHAR.

Esempi

L'esempio seguente riduce gli spazi vuoti iniziali e finali dalla stringa `' abc '`:

```
select '     abc    ' as untrim, btrim('     abc    ') as trim;

untrim    | trim
----------+------
   abc    | abc
```

L'esempio seguente rimuove le stringhe `'xyz'` iniziali e finali dalla stringa `'xyzaxyzbxyzcxyz'`. Le occorrenze iniziali e finali di `'xyz'` vengono rimosse, ma le occorrenze interne alla stringa non vengono rimosse.

```
select 'xyzaxyzbxyzcxyz' as untrim,
btrim('xyzaxyzbxyzcxyz', 'xyz') as trim;

     untrim      |   trim
-----------------+-----------
 xyzaxyzbxyzcxyz | axyzbxyzc
```

L'esempio seguente rimuove le parti iniziale e finale dalla stringa `'setuphistorycassettes'` che corrispondono a uno qualsiasi dei caratteri nell'elenco `'tes'` trim_chars. Qualsiasi carattere `t`, `e` o `s` che si verifica prima di un altro carattere che non è nell'elenco trim_chars all'inizio o alla fine della stringa di input viene rimosso.

```
SELECT btrim('setuphistorycassettes', 'tes');

     btrim
-----------------
 uphistoryca
```

Funzione BTTEXT_PATTERN_CMP

Sinonimo della funzione BPCHARCMP.

Per informazioni dettagliate, consultare Funzione BPCHARCMP.

Funzione CHAR_LENGTH

Sinonimo della funzione LEN.

Per informazioni, consulta Funzione LEN.

Funzione CHARACTER_LENGTH

Sinonimo della funzione LEN.

Per informazioni, consulta Funzione LEN.

Funzione CHARINDEX

Restituisce la posizione della sottostringa specificata all'interno di una stringa.

Per funzioni simili, consulta Funzione POSITION e Funzione STRPOS.

Sintassi

```
CHARINDEX( substring, string )
```

Argomenti

sottostringa

La sottostringa da cercare all'interno della stringa.

stringa

La stringa o colonna da ricercare.

Tipo restituito

INTEGER

La funzione CHARINDEX restituisce un valore `INTEGER` corrispondente alla posizione della sottostringa (basata su uno, non su zero). La posizione si basa sul numero di caratteri, non di

byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. CHARINDEX restituisce `0` se la sottostringa non si trova all'interno della stringa.

Esempi

Per restituire la posizione della stringa `fish` all'interno della parola dog, utilizza l'esempio seguente.

```
SELECT CHARINDEX('fish', 'dog');

+-----------+
| charindex |
+-----------+
|         0 |
+-----------+
```

Per restituire la posizione della stringa `fish` all'interno della parola `dogfish`, utilizza l'esempio seguente.

```
SELECT CHARINDEX('fish', 'dogfish');

+-----------+
| charindex |
+-----------+
|         4 |
+-----------+
```

Nell'esempio seguente viene utilizzata la tabella SALES del database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire il numero di transazioni di vendita distinte con una commissione superiore a 999,00 dalla tabella SALES, utilizza l'esempio seguente. Questo comando conta le commissioni superiori a 999,00 controllando se il valore decimale è superiore a 4 posizioni dall'inizio del valore della commissione.

```
SELECT DISTINCT CHARINDEX('.', commission), COUNT (CHARINDEX('.', commission))
FROM sales
WHERE CHARINDEX('.', commission) > 4
GROUP BY CHARINDEX('.', commission)
ORDER BY 1,2;
```

```
+-----------+-------+
| charindex | count |
+-----------+-------+
|         5 |   629 |
+-----------+-------+
```

Funzione CHR

La funzione CHR restituisce il carattere che corrisponde al valore del punto di codice ASCII specificato dal parametro di input.

Sintassi

```
CHR(number)
```

Argomento

numero

Il parametro di input è un valore `INTEGER` che rappresenta un valore del punto di codice ASCII.

Tipo restituito

CHAR

La funzione CHR restituisce una stringa `CHAR` se un carattere ASCII corrisponde al valore di input. Se il numero di input non ha corrispondenza ASCII, la funzione restituisce `NULL`.

Esempi

Per restituire il carattere che corrisponde al punto 0 del codice ASCII, utilizza l'esempio seguente. La funzione CHR restituisce `NULL` per l'input `0`.

```
SELECT CHR(0);

+-----+
| chr |
+-----+
|     |
+-----+
```

Per restituire il carattere che corrisponde al punto 65 del codice ASCII, utilizza l'esempio seguente.

```
SELECT CHR(65);

+-----+
| chr |
+-----+
| A   |
+-----+
```

Per restituire nomi di eventi distinti che iniziano con la A maiuscola (punto 65 del codice ASCII), utilizza l'esempio seguente. Nell'esempio seguente viene utilizzata la tabella EVENT del database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

```
SELECT DISTINCT eventname FROM event
WHERE SUBSTRING(eventname, 1, 1)=CHR(65) LIMIT 5;

+----------------------+
|       eventname      |
+----------------------+
| A Catered Affair     |
| As You Like It       |
| A Man For All Seasons |
| Alan Jackson         |
| Armando Manzanero    |
+----------------------+
```

Funzione COLLATE

La funzione COLLATE sovrascrive il confronto di una colonna o di un'espressione stringa.

Per informazioni su come creare le tabelle mediante il confronto di database, consultare CREATE TABLE.

Per informazioni su come creare i database mediante il confronto di database, consultare CREATE DATABASE.

Sintassi

```
COLLATE( string, 'case_sensitive' | 'case_insensitive');
```

Argomenti

stringa

Una colonna o un'espressione di stringa che desideri sovrascrivere.

'case_sensitive' | 'case_insensitive'

Una costante di stringa di un nome di confronto. Amazon Redshift supporta solo case_sensitive o case_insensitive.

Tipo restituito

La funzione COLLATE restituisce VARCHAR o CHAR a seconda del primo tipo di espressione di input. Questa funzione modifica il confronto solo del primo argomento di input e non ne modifica il valore di output.

Esempi

Per creare la tabella T e definire col1 nella tabella T come `case_sensitive`, utilizza il seguente esempio.

```
CREATE TABLE T ( col1 Varchar(20) COLLATE case_sensitive );

INSERT INTO T VALUES ('john'),('JOHN');
```

Quando esegui la prima query, Amazon Redshift restituisce solo `john`. Dopo che la funzione COLLATE viene eseguita su col1, il confronto diventa `case_insensitive`. La seconda query restituisce sia `john` che `JOHN`.

```
SELECT * FROM T WHERE col1 = 'john';

+------+
| col1 |
+------+
| john |
+------+

SELECT * FROM T WHERE COLLATE(col1, 'case_insensitive') = 'john';

+------+
```

```
| col1 |
+------+
| john |
| JOHN |
+------+
```

Per creare la tabella A e definire col1 nella tabella A come `case_insensitive`, utilizza il seguente esempio.

```
CREATE TABLE A ( col1 Varchar(20) COLLATE case_insensitive );

INSERT INTO A VALUES ('john'),('JOHN');
```

Quando esegui la prima query, Amazon Redshift restituisce sia `john` che `JOHN`. Dopo che la funzione COLLATE viene eseguita su col1, il confronto diventa `case_sensitive`. La seconda query restituisce solo `john`.

```
SELECT * FROM A WHERE col1 = 'john';

+------+
| col1 |
+------+
| john |
| JOHN |
+------+

SELECT * FROM A WHERE COLLATE(col1, 'case_sensitive') = 'john';

+------+
| col1 |
+------+
| john |
+------+
```

Funzione CONCAT

La funzione CONCAT concatena due espressioni e restituisce l'espressione risultante. Per concatenare più di due espressioni, utilizzare le funzioni CONCAT nidificate. L'operatore di concatenazione (`||`) tra due espressioni produce gli stessi risultati della funzione CONCAT.

Sintassi

```
CONCAT ( expression1, expression2 )
```

Argomenti

expression1, expression2

Entrambi gli argomenti possono essere una stringa di caratteri a lunghezza fissa, una stringa di caratteri a lunghezza variabile, un'espressione binaria o un'espressione che valuta uno di questi input.

Tipo restituito

CONCAT restituisce un'espressione. Il tipo di dati dell'espressione è lo stesso tipo degli argomenti di input.

Se le espressioni di input sono di tipo diverso, Amazon Redshift tenta di digitare implicitamente una delle espressioni. Se non è possibile eseguire il cast di valori, viene restituito il valore nullo.

Note per l'utilizzo

- Sia per la funzione CONCAT sia per l'operatore di concatenazione, se una o entrambe le espressioni sono nulle, il risultato della concatenazione è nullo.

Esempi

L'esempio seguente concatena due letterali di caratteri:

```
SELECT CONCAT('December 25, ', '2008');

concat
-------------------
December 25, 2008
(1 row)
```

La seguente query, utilizzando l'operatore || invece di CONCAT, produce lo stesso risultato:

```
SELECT 'December 25, '||'2008';
```

```
?column?
-------------------
December 25, 2008
(1 row)
```

L'esempio seguente utilizza una funzione CONCAT nidificata all'interno di un'altra funzione CONCAT per concatenare tre stringhe di caratteri:

```
SELECT CONCAT('Thursday, ', CONCAT('December 25, ', '2008'));

concat
----------------------------
Thursday, December 25, 2008
(1 row)
```

Per concatenare colonne che potrebbero contenere valori NULL, usa Funzioni NVL e COALESCE, che restituisce un determinato valore quando rileva un valore NULL. Il seguente esempio utilizza NVL per restituire uno 0 ogni volta che si incontra NULL.

```
SELECT CONCAT(venuename, CONCAT(' seats ', NVL(venueseats, 0))) AS seating
FROM venue WHERE venuestate = 'NV' OR venuestate = 'NC'
ORDER BY 1
LIMIT 5;

seating
-----------------------------------
Ballys Hotel seats 0
Bank of America Stadium seats 73298
Bellagio Hotel seats 0
Caesars Palace seats 0
Harrahs Hotel seats 0
(5 rows)
```

La query seguente concatena i valori CITY e STATE dalla tabella VENUE:

```
SELECT CONCAT(venuecity, venuestate)
FROM venue
WHERE venueseats > 75000
ORDER BY venueseats;

concat
-------------------
```

```
DenverCO
Kansas CityMO
East RutherfordNJ
LandoverMD
(4 rows)
```

La seguente query utilizza funzioni CONCAT nidificate. La query concatena i valori CITY e STATE dalla tabella VENUE ma delimita la stringa risultante con una virgola e uno spazio:

```
SELECT CONCAT(CONCAT(venuecity,', '),venuestate)
FROM venue
WHERE venueseats > 75000
ORDER BY venueseats;

concat
---------------------
Denver, CO
Kansas City, MO
East Rutherford, NJ
Landover, MD
(4 rows)
```

L'esempio seguente concatena due espressioni binarie. Dove `abc` è un valore binario (con una rappresentazione esadecimale di `616263`) e `def` è un valore binario (con rappresentazione esadecimale di `646566`). Il risultato viene visualizzato automaticamente come rappresentazione esadecimale del valore binario.

```
SELECT CONCAT('abc'::VARBYTE, 'def'::VARBYTE);

concat
-------------------
616263646566
```

Funzione CRC32

CRC32 è una funzione utilizzata per il rilevamento degli errori. La funzione un algoritmo CRC32 per rilevare le modifiche tra i dati di origine e di destinazione. La funzione CRC32 converte una stringa di lunghezza variabile in una stringa di 8 caratteri che è una rappresentazione testuale del valore esadecimale di una sequenza binaria a 32 bit. Per rilevare le modifiche tra i dati di origine e di destinazione, utilizza la funzione CRC32 sui dati di origine e archivia l'output. Quindi, utilizza la

funzione CRC32 sui dati di destinazione e confronta l'output con l'output dei dati di origine. Se i dati non sono stati modificati, gli output saranno gli stessi, altrimenti, gli output saranno diversi.

Sintassi

```
CRC32(string)
```

Argomenti

stringa

Una stringa CHAR, una stringa VARCHAR o un'espressione che restituisce implicitamente un tipo CHAR o VARCHAR.

Tipo restituito

La funzione CRC32 restituisce una stringa di 8 caratteri che è una rappresentazione testuale del valore esadecimale di una sequenza binaria a 32 bit. La funzione CRC32 di Amazon Redshift si basa sul polinomio CRC-32C.

Esempi

Mostrare il valore a 8 bit per la stringa `Amazon Redshift`.

```
SELECT CRC32('Amazon Redshift');

+----------+
|  crc32   |
+----------+
| f2726906 |
+----------+
```

Funzione DIFFERENCE

La funzione DIFFERENCE confronta i codici American Soundex di due stringhe. La funzione restituisce un valore `INTEGER` per indicare il numero di caratteri corrispondenti tra i codici Soundex.

Un codice Soundex è una stringa lunga quattro caratteri. Un codice Soundex rappresenta la fonetica di una parola anziché il modo in cui viene scritta. Ad esempio `Smith` e `Smyth` hanno lo stesso codice Soundex.

Sintassi

```
DIFFERENCE(string1, string2)
```

Argomenti

string1

Una stringa CHAR, una stringa VARCHAR o un'espressione che restituisce implicitamente un tipo CHAR o VARCHAR.

string2

Una stringa CHAR, una stringa VARCHAR o un'espressione che restituisce implicitamente un tipo CHAR o VARCHAR.

Tipo restituito

INTEGER

La funzione DIFFERENCE restituisce un valore INTEGER da 0 a 4 che conta il numero di caratteri corrispondenti nei codici American Soundex delle due stringhe. Un codice Soundex ha 4 caratteri, quindi la funzione DIFFERENCE restituisce 4 quando tutti e 4 i caratteri dei valori del codice American Soundex delle stringhe sono uguali. DIFFERENCE restituisce 0 se una delle due stringhe è vuota. La funzione restituisce 1 se nessuna stringa contiene caratteri validi. La funzione DIFFERENCE converte solo caratteri ASCII alfabetici minuscoli o maiuscoli inglesi, inclusi a-z e A-Z. DIFFERENCE ignora gli altri caratteri.

Esempi

Per confrontare i valori Soundex delle stringhe % e @, utilizza l'esempio seguente. La funzione restituisce 1 poiché nessuna stringa contiene caratteri validi.

```
SELECT DIFFERENCE('%', '@');

+------------+
| difference |
+------------+
|          1 |
+------------+
```

Per confrontare i valori Soundex di `Amazon` e di una stringa vuota, utilizza l'esempio seguente. La funzione restituisce `0` poiché una delle due stringhe è vuota.

```
SELECT DIFFERENCE('Amazon', '');

+------------+
| difference |
+------------+
|          0 |
+------------+
```

Per confrontare i valori Soundex delle stringhe `Amazon` e `Ama`, utilizza l'esempio seguente. La funzione restituisce 2 perché 2 caratteri dei valori Soundex delle stringhe sono uguali.

```
SELECT DIFFERENCE('Amazon', 'Ama');

+------------+
| difference |
+------------+
|          2 |
+------------+
```

Per confrontare i valori Soundex delle stringhe `Amazon` e `+-*/%Amazon`, utilizza l'esempio seguente. La funzione restituisce `4` perché tutti e 4 i caratteri dei valori Soundex delle stringhe sono uguali. Tieni presente che la funzione ignora i caratteri `+-*/%` non validi nella seconda stringa.

```
SELECT DIFFERENCE('Amazon', '+-*/%Amazon');

+------------+
| difference |
+------------+
|          4 |
+------------+
```

Per confrontare i valori Soundex delle stringhe `AC/DC` e `Ay See Dee See`, utilizza l'esempio seguente. La funzione restituisce `4` perché tutti e 4 i caratteri dei valori Soundex delle stringhe sono uguali.

```
SELECT DIFFERENCE('AC/DC', 'Ay See Dee See');
```

```
+------------+
| difference |
+------------+
|          4 |
+------------+
```

Funzione INITCAP

Scrive con la lettera maiuscola la prima lettera di ogni parola in una stringa specificata. INITCAP supporta caratteri multibyte UTF-8, fino a un massimo di quattro byte per carattere.

Sintassi

```
INITCAP(string)
```

Argomento

stringa

Una stringa CHAR, una stringa VARCHAR o un'espressione che restituisce implicitamente un tipo CHAR o VARCHAR.

Tipo restituito

VARCHAR

Note per l'utilizzo

La funzione INITCAP rende maiuscola la prima lettera di ogni parola in una stringa e rende (o lascia) minuscole tutte le lettere successive. Pertanto, è importante capire quali caratteri (diversi dai caratteri di spazio) fungono da separatori di parole. Un carattere separatore di parole è un carattere non alfanumerico, compresi i segni di punteggiatura, i simboli e i caratteri di controllo. Tutti i seguenti caratteri sono separatori di parole:

```
! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~
```

Le schede, i caratteri di nuova riga, i feed del modulo, i feed di riga e i rimandi a capo sono anch'essi separatori di parole.

Esempi

Gli esempi seguenti utilizzano i dati delle tabelle CATEGORY e USERS dal database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per scrivere in maiuscolo le iniziali di ciascuna parola nella colonna CATDESC, utilizza l'esempio seguente.

```
SELECT catid, catdesc, INITCAP(catdesc)
FROM category
ORDER BY 1, 2, 3;

+-------+----------------------------------------
+----------------------------------------+
| catid |                 catdesc                 |                 initcap
          |
+-------+----------------------------------------
+----------------------------------------+
|     1 | Major League Baseball                   | Major League Baseball
          |
|     2 | National Hockey League                  | National Hockey League
          |
|     3 | National Football League                | National Football League
          |
|     4 | National Basketball Association         | National Basketball Association
          |
|     5 | Major League Soccer                     | Major League Soccer
          |
|     6 | Musical theatre                         | Musical Theatre
          |
|     7 | All non-musical theatre                 | All Non-Musical Theatre
          |
|     8 | All opera and light opera               | All Opera And Light Opera
          |
|     9 | All rock and pop music concerts         | All Rock And Pop Music Concerts
          |
|    10 | All jazz singers and bands              | All Jazz Singers And Bands
          |
|    11 | All symphony, concerto, and choir concerts | All Symphony, Concerto, And
 Choir Concerts |
+-------+----------------------------------------
+----------------------------------------+
```

Per mostrare che la funzione INITCAP non conserva i caratteri maiuscoli quando non si trovano all'inizio delle parole, utilizza l'esempio seguente. Ad esempio, la stringa `MLB` diventa `Mlb`.

```
SELECT INITCAP(catname)
FROM category
ORDER BY catname;

+-----------+
|  initcap  |
+-----------+
| Classical |
| Jazz      |
| Mlb       |
| Mls       |
| Musicals  |
| Nba       |
| Nfl       |
| Nhl       |
| Opera     |
| Plays     |
| Pop       |
+-----------+
```

Per mostrare che i caratteri non alfanumerici diversi dagli spazi funzionano come separatori di parole, utilizza l'esempio seguente. Diverse lettere di ogni stringa verranno scritte in maiuscolo.

```
SELECT email, INITCAP(email)
FROM users
ORDER BY userid DESC LIMIT 5;

+------------------------------------+------------------------------------+
|               email                |              initcap               |
+------------------------------------+------------------------------------+
| urna.Ut@egetdictumplacerat.edu     | Urna.Ut@Egetdictumplacerat.Edu     |
| nibh.enim@egestas.ca               | Nibh.Enim@Egestas.Ca               |
| in@Donecat.ca                      | In@Donecat.Ca                      |
| sodales@blanditviverraDonec.ca     | Sodales@Blanditviverradonec.Ca     |
| sociis.natoque.penatibus@vitae.org | Sociis.Natoque.Penatibus@Vitae.Org |
+------------------------------------+------------------------------------+
```

Funzioni LEFT e RIGHT

Queste funzioni restituiscono il numero specificato di caratteri più a sinistra o più a destra da una stringa di caratteri.

Il numero si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli.

Sintassi

```
LEFT( string,  integer )

RIGHT( string,  integer )
```

Argomenti

stringa

Una stringa CHAR, una stringa VARCHAR o un'espressione che restituisce una stringa CHAR o VARCHAR.

integer

Un integer positivo.

Tipo restituito

VARCHAR

Esempi

Nell'esempio seguente vengono utilizzati i dati della tabella EVENT del database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire i 5 caratteri più a sinistra e i 5 caratteri più a destra dai nomi di eventi che hanno ID compresi tra 1000 e 1005, utilizza l'esempio seguente.

```
SELECT eventid, eventname,
LEFT(eventname,5) AS left_5,
RIGHT(eventname,5) AS right_5
FROM event
WHERE eventid BETWEEN 1000 AND 1005
ORDER BY 1;
```

```
+---------+----------------+--------+---------+
| eventid |   eventname    | left_5 | right_5 |
+---------+----------------+--------+---------+
|    1000 | Gypsy          | Gypsy  | Gypsy   |
|    1001 | Chicago        | Chica  | icago   |
|    1002 | The King and I | The K  | and I   |
|    1003 | Pal Joey       | Pal J  | Joey    |
|    1004 | Grease         | Greas  | rease   |
|    1005 | Chicago        | Chica  | icago   |
+---------+----------------+--------+---------+
```

Funzione LEN

Restituisce la lunghezza della stringa specificata come numero di caratteri.

Sintassi

LEN è un sinonimo di Funzione LENGTH, Funzione CHAR_LENGTH, Funzione CHARACTER_LENGTH, e Funzione TEXTLEN.

```
LEN(expression)
```

Argomento

espressione

Una stringa CHAR, una stringa VARCHAR, un'espressione VARBYTE o un'espressione che restituisce implicitamente un tipo CHAR, VARCHAR o VARBYTE.

Tipo restituito

INTEGER

La funzione LEN restituisce un integer che indica il numero di caratteri nella stringa di input.

Se la stringa di input è una stringa di caratteri, la funzione LEN restituisce il numero effettivo di caratteri nelle stringhe multi-byte, non il numero di byte. Ad esempio, è necessaria una colonna VARCHAR(12) per archiviare tre caratteri cinesi a quattro byte. La funzione LEN restituirà 3 per quella stessa stringa. Per ottenere la lunghezza di una stringa in byte, utilizzare la funzione OCTET_LENGTH.

Note per l'utilizzo

Se expression è una stringa CHAR, gli spazi finali non vengono contati.

Se expression è una stringa VARCHAR, gli spazi finali vengono contati.

Esempi

Per restituire il numero di byte e il numero di caratteri nella stringa `français`, utilizza l'esempio seguente.

```
SELECT OCTET_LENGTH('français'),
LEN('français');

+--------------+-----+
| octet_length | len |
+--------------+-----+
|            9 |   8 |
+--------------+-----+
```

Per restituire il numero di byte e il numero di caratteri nella stringa `français` senza utilizzare la funzione OCTET_LENGTH, utilizza l'esempio seguente. Per ulteriori informazioni, consulta Funzione CAST.

```
SELECT LEN(CAST('français' AS VARBYTE)) as bytes, LEN('français');

+-------+-----+
| bytes | len |
+-------+-----+
|     9 |   8 |
+-------+-----+
```

Per restituire il numero di caratteri nelle stringhe `cat` senza spazi finali, `cat  ` con tre spazi finali, `cat` con conversione di tre spazi finali come CHAR di lunghezza 6 e `cat  ` con conversione di tre spazi finali come VARCHAR di lunghezza 6, utilizza l'esempio seguente. Tieni presente che la funzione non conta gli spazi finali per le stringhe CHAR, ma conta gli spazi finali per le stringhe VARCHAR.

```
SELECT LEN('cat'), LEN('cat   '), LEN(CAST('cat   ' AS CHAR(6))) AS len_char,
 LEN(CAST('cat   ' AS VARCHAR(6))) AS len_varchar;
```

```
+-----+-----+----------+-------------+
| len | len | len_char | len_varchar |
+-----+-----+----------+-------------+
|   3 |   6 |        3 |           6 |
+-----+-----+----------+-------------+
```

Nell'esempio seguente vengono utilizzati i dati della tabella VENUE database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire i dieci nomi VENUE più lunghi nella tabella VENUE, utilizza l'esempio seguente.

```
SELECT venuename, LEN(venuename)
FROM venue
ORDER BY 2 DESC, 1
LIMIT 10;

+----------------------------------------+-----+
|               venuename                | len |
+----------------------------------------+-----+
| Saratoga Springs Performing Arts Center |  39 |
| Lincoln Center for the Performing Arts  |  38 |
| Nassau Veterans Memorial Coliseum       |  33 |
| Jacksonville Municipal Stadium          |  30 |
| Rangers BallPark in Arlington           |  29 |
| University of Phoenix Stadium           |  29 |
| Circle in the Square Theatre            |  28 |
| Hubert H. Humphrey Metrodome            |  28 |
| Oriole Park at Camden Yards             |  27 |
| Dick's Sporting Goods Park              |  26 |
+----------------------------------------+-----+
```

Funzione LENGTH

Sinonimo della funzione LEN.

Per informazioni, consulta Funzione LEN.

Funzione LOWER

Converte una stringa in minuscolo. LOWER supporta caratteri multibyte UTF-8, fino a un massimo di quattro byte per carattere.

Sintassi

```
LOWER(string)
```

Argomento

stringa

Una stringa VARCHAR o qualsiasi espressione che restituisce il tipo VARCHAR.

Tipo restituito

string

La funzione LOWER restituisce una stringa che appartiene allo stesso tipo di dati della stringa di input. Ad esempio, se l'input è una stringa CHAR, la funzione restituirà una stringa CHAR.

Esempi

Nell'esempio seguente vengono utilizzati i dati della tabella CATEGORY database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per convertire le stringhe VARCHAR nella colonna CATNAME in lettere minuscole, utilizza l'esempio seguente.

```
SELECT catname, LOWER(catname) FROM category ORDER BY 1,2;

+-----------+-----------+
|  catname  |   lower   |
+-----------+-----------+
| Classical | classical |
| Jazz      | jazz      |
| MLB       | mlb       |
| MLS       | mls       |
| Musicals  | musicals  |
| NBA       | nba       |
| NFL       | nfl       |
| NHL       | nhl       |
| Opera     | opera     |
| Plays     | plays     |
| Pop       | pop       |
```

```
+-----------+-----------+
```

Funzioni LPAD e RPAD

Queste funzioni antepongono o aggiungono caratteri a una stringa, in base a una lunghezza specificata.

Sintassi

```
LPAD(string1, length, [ string2 ])
```

```
RPAD(string1, length, [ string2 ])
```

Argomenti

string1

Una stringa CHAR, una stringa VARCHAR o un'espressione che restituisce implicitamente un tipo CHAR o VARCHAR.

length

Un integer che definisce la lunghezza del risultato della funzione. La lunghezza di una stringa si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. Se string1 è più lunga della lunghezza specificata, viene troncata (a destra). Se length è un numero negativo, il risultato della funzione è una stringa vuota.

string2

(Facoltativo) Uno o più caratteri anteposti o aggiunti a string1. Questo argomento non è specificato, vengono usati gli spazi.

Tipo restituito

VARCHAR

Esempi

Negli esempi seguenti vengono utilizzati i dati della tabella EVENT del database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per troncare un set specificato di nomi di eventi a 20 caratteri e anteporre ai nomi più brevi gli spazi, utilizza l'esempio seguente.

```
SELECT LPAD(eventname, 20) FROM event
WHERE eventid BETWEEN 1 AND 5 ORDER BY 1;

+----------------------+
|         lpad         |
+----------------------+
|               Salome |
|        Il Trovatore |
|       Boris Godunov |
|     Gotterdammerung |
|La Cenerentola (Cind |
+----------------------+
```

Per troncare lo stesso set di nomi di eventi a 20 caratteri ma aggiungere ai nomi più brevi `0123456789`, utilizza l'esempio seguente.

```
SELECT RPAD(eventname, 20,'0123456789') FROM event
WHERE eventid BETWEEN 1 AND 5 ORDER BY 1;

+----------------------+
|         rpad         |
+----------------------+
| Boris Godunov0123456 |
| Gotterdammerung01234 |
| Il Trovatore01234567 |
| La Cenerentola (Cind |
| Salome01234567890123 |
+----------------------+
```

Funzione LTRIM

Taglia i caratteri dall'inizio di una stringa. Rimuove la stringa più lunga contenente solo i caratteri nell'elenco dei caratteri di taglio. Il taglio è completo quando un carattere di taglio non appare nella stringa di input.

Sintassi

```
LTRIM( string [, trim_chars] )
```

Argomenti

stringa

Una stringa, una colonna, un'espressione o una stringa letterale da tagliare.

trim_chars

Una colonna o un'espressione di stringhe o un valore letterale di stringa che rappresenta i caratteri da tagliare dall'inizio della stringa. Se non specificato, viene utilizzato uno spazio come carattere di taglio.

Tipo restituito

La funzione LTRIM restituisce una stringa di caratteri che appartiene allo stesso tipo di dati della stringa di input (CHAR o VARCHAR).

Esempi

L'esempio seguente taglia l'anno dalla colonna `listime`. I caratteri di taglio nel valore letterale di stringa `'2008-'` indicano i caratteri da tagliare da sinistra. Se si utilizzano i caratteri di taglio `'028-'`, si ottiene lo stesso risultato.

```
select listid, listtime, ltrim(listtime, '2008-')
from listing
order by 1, 2, 3
limit 10;

listid |      listtime       |     ltrim
-------+---------------------+---------------
     1 | 2008-01-24 06:43:29 | 1-24 06:43:29
     2 | 2008-03-05 12:25:29 | 3-05 12:25:29
     3 | 2008-11-01 07:35:33 | 11-01 07:35:33
     4 | 2008-05-24 01:18:37 | 5-24 01:18:37
     5 | 2008-05-17 02:29:11 | 5-17 02:29:11
     6 | 2008-08-15 02:08:13 | 15 02:08:13
     7 | 2008-11-15 09:38:15 | 11-15 09:38:15
     8 | 2008-11-09 05:07:30 | 11-09 05:07:30
     9 | 2008-09-09 08:03:36 | 9-09 08:03:36
    10 | 2008-06-17 09:44:54 | 6-17 09:44:54
```

LTRIM rimuove tutti i caratteri in trim_chars se questi si trovano all'inizio di stringa. L'esempio seguente riduce i caratteri "C", "D" e "G" quando si trovano all'inizio di VENUENAME che è una colonna VARCHAR.

```
select venueid, venuename, ltrim(venuename, 'CDG')
from venue
where venuename like '%Park'
order by 2
limit 7;

venueid | venuename                  | btrim
--------+----------------------------+-------------------------
    121 | ATT Park                   | ATT Park
    109 | Citizens Bank Park         | itizens Bank Park
    102 | Comerica Park              | omerica Park
      9 | Dick's Sporting Goods Park | ick's Sporting Goods Park
     97 | Fenway Park                | Fenway Park
    112 | Great American Ball Park   | reat American Ball Park
    114 | Miller Park                | Miller Park
```

L'esempio seguente utilizza il carattere di taglio 2 che viene recuperato dalla colonna `venueid`.

```
select ltrim('2008-01-24 06:43:29', venueid)
from venue where venueid=2;

ltrim
------------------
008-01-24 06:43:29
```

L'esempio seguente non taglia alcun carattere perché prima del carattere di taglio `'0'` è presente un 2.

```
select ltrim('2008-01-24 06:43:29', '0');

ltrim
-------------------
2008-01-24 06:43:29
```

L'esempio seguente utilizza il carattere di taglio dello spazio predefinito e taglia i due spazi dall'inizio della stringa.

```
select ltrim('  2008-01-24 06:43:29');

ltrim
-------------------
2008-01-24 06:43:29
```

Funzione OCTETINDEX

La funzione OCTETINDEX restituisce la posizione di una sottostringa all'interno di una stringa come un numero di byte.

Sintassi

```
OCTETINDEX(substring, string)
```

Argomenti

sottostringa

Una stringa CHAR, una stringa VARCHAR o un'espressione che restituisce implicitamente un tipo CHAR o VARCHAR.

stringa

Una stringa CHAR, una stringa VARCHAR o un'espressione che restituisce implicitamente un tipo CHAR o VARCHAR.

Tipo restituito

INTEGER

La funzione OCTETINDEX restituisce un valore INTEGER corrispondente alla posizione della sottostringa all'interno della stringa come un numero di byte, in cui il primo carattere della stringa viene conteggiato come 1. Se la stringa non contiene caratteri multibyte, il risultato è uguale al risultato della funzione CHARINDEX. Se la stringa non contiene la sottostringa, la funzione restituisce 0. Se la sottostringa è vuota la funzione restituisce 1.

Esempi

Per restituire la posizione della sottostringa q all'interno della stringa `Amazon Redshift`, utilizza l'esempio seguente. Questo esempio restituisce `0` perché la sottostringa non è nella stringa.

```
SELECT OCTETINDEX('q', 'Amazon Redshift');

+------------+
| octetindex |
+------------+
|          0 |
+------------+
```

Per restituire la posizione di una sottostringa vuota all'interno della stringa `Amazon Redshift`, utilizza l'esempio seguente. Questo esempio restituisce 1 perché la sottostringa è vuota.

```
SELECT OCTETINDEX('', 'Amazon Redshift');

+------------+
| octetindex |
+------------+
|          1 |
+------------+
```

Per restituire la posizione della sottostringa `Redshift` all'interno della stringa `Amazon Redshift`, utilizza l'esempio seguente. Nell'esempio seguente viene restituito 8 perché la sottostringa inizia dall'ottavo byte della stringa.

```
SELECT OCTETINDEX('Redshift', 'Amazon Redshift');

+------------+
| octetindex |
+------------+
|          8 |
+------------+
```

Per restituire la posizione della sottostringa `Redshift` all'interno della stringa `Amazon Redshift`, utilizza l'esempio seguente. Questo esempio restituisce 21 perché i primi sei caratteri della stringa sono caratteri a doppio byte.

```
SELECT OCTETINDEX('Redshift', 'Άμαζον Amazon Redshift');
```

```
+------------+
| octetindex |
+------------+
|         21 |
+------------+
```

Funzione OCTET_LENGTH

Restituisce la lunghezza della stringa specificata come numero di byte.

Sintassi

```
OCTET_LENGTH(expression)
```

Argomento

espressione

Una stringa CHAR, una stringa VARCHAR, un'espressione VARBYTE o un'espressione che restituisce implicitamente un tipo CHAR, VARCHAR o VARBYTE.

Tipo restituito

INTEGER

La funzione OCTET_LENGTH restituisce un integer che indica il numero di byte nella stringa di input.

Se la stringa di input è una stringa di caratteri, la funzione LEN restituisce il numero effettivo di caratteri nelle stringhe multi-byte, non il numero di byte. Ad esempio, è necessaria una colonna VARCHAR(12) per archiviare tre caratteri cinesi a quattro byte. La funzione OCTET_LENGTH restituirà 12 per quella stringa e la funzione LEN restituirà 3 per la stessa stringa.

Note per l'utilizzo

Se expression è una stringa CHAR, la funzione restituisce la lunghezza della stringa CHAR. Ad esempio, l'output di un input CHAR(6) è CHAR(6).

Se expression è una stringa VARCHAR, gli spazi finali vengono contati.

Esempi

Per restituire il numero di byte quando la stringa `francais` con tre spazi finali viene convertita in un tipo CHAR e VARCHAR, utilizza l'esempio seguente. Per ulteriori informazioni, consulta Funzione CAST.

```
SELECT OCTET_LENGTH(CAST('francais   ' AS CHAR(15))) AS octet_length_char,
 OCTET_LENGTH(CAST('francais   ' AS VARCHAR(15))) AS octet_length_varchar;

+------------------+----------------------+
| octet_length_char | octet_length_varchar |
+------------------+----------------------+
|                15 |                   11 |
+------------------+----------------------+
```

Per restituire il numero di byte e il numero di caratteri nella stringa `français`, utilizza l'esempio seguente.

```
SELECT OCTET_LENGTH('français'), LEN('français');

+--------------+-----+
| octet_length | len |
+--------------+-----+
|            9 |   8 |
+--------------+-----+
```

Per restituire il numero di byte quando la stringa `français` viene convertita in VARBYTE, utilizza l'esempio seguente.

```
SELECT OCTET_LENGTH(CAST('français' AS VARBYTE));

+--------------+
| octet_length |
+--------------+
|            9 |
+--------------+
```

Funzione POSITION

Restituisce la posizione della sottostringa specificata all'interno di una stringa.

Per funzioni simili, consulta Funzione CHARINDEX e Funzione STRPOS.

Sintassi

```
POSITION(substring IN string )
```

Argomenti

sottostringa

La sottostringa da cercare all'interno della stringa.

stringa

La stringa o colonna da ricercare.

Tipo restituito

La funzione POSITION restituisce un valore `INTEGER` corrispondente alla posizione della sottostringa (basata su uno, non su zero). La posizione si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. POSITION restituisce `0` se la sottostringa non si trova all'interno della stringa.

Esempi

Per restituire la posizione della stringa `fish` all'interno della parola `dog`, utilizza l'esempio seguente.

```
SELECT POSITION('fish' IN 'dog');

+-----------+
|  position |
+-----------+
|         0 |
+-----------+
```

Per restituire la posizione della stringa `fish` all'interno della parola `dogfish`, utilizza l'esempio seguente.

```
SELECT POSITION('fish' IN 'dogfish');

+-----------+
|  position |
+-----------+
|         4 |
```

```
+-----------+
```

Nell'esempio seguente viene utilizzata la tabella SALES del database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire il numero di transazioni di vendita distinte con una commissione superiore a 999,00 dalla tabella SALES, utilizza l'esempio seguente. Questo comando conta le commissioni superiori a 999,00 controllando se il valore decimale è superiore a 4 posizioni dall'inizio del valore della commissione.

```
SELECT DISTINCT POSITION('.' IN commission), COUNT (POSITION('.' IN commission))
FROM sales
WHERE POSITION('.' IN commission) > 4
GROUP BY POSITION('.' IN commission)
ORDER BY 1,2;

+-----------+-------+
|  position | count |
+-----------+-------+
|         5 |   629 |
+-----------+-------+
```

Funzione QUOTE_IDENT

La funzione QUOTE_IDENT restituisce la stringa specificata come una stringa con virgolette doppie iniziali e virgolette doppie finali. L'output della funzione può essere utilizzato come identificatore in un'istruzione SQL. La funzione raddoppia in modo appropriato qualsiasi virgoletta doppia incorporata.

QUOTE_IDENT aggiunge le doppie virgolette solo quando è necessario per creare un identificatore valido, quando la stringa contiene caratteri non identificativi o sarebbe altrimenti espressa in minuscolo. Per restituire sempre una stringa con virgoletta singola, utilizzare QUOTE_LITERAL.

Sintassi

```
QUOTE_IDENT(string)
```

Argomento

stringa

Una stringa CHAR o VARCHAR.

Tipo restituito

La funzione QUOTE_IDENT restituisce lo stesso tipo di stringa della stringa di input.

Esempi

Per restituire la stringa "CAT" con virgolette doppie, utilizza l'esempio seguente.

```
SELECT QUOTE_IDENT('"CAT"');

+-------------+
| quote_ident |
+-------------+
| """CAT"""   |
+-------------+
```

Nell'esempio seguente vengono utilizzati i dati della tabella CATEGORY del database TICKIT di esempio. Per ulteriori informazioni, consulta Database di esempio.

Per restituire la colonna CATNAME racchiusa tra virgolette, utilizza l'esempio seguente.

```
SELECT catid, QUOTE_IDENT(catname)
FROM category
ORDER BY 1,2;

+-------+-------------+
| catid | quote_ident |
+-------+-------------+
|     1 | "MLB"       |
|     2 | "NHL"       |
|     3 | "NFL"       |
|     4 | "NBA"       |
|     5 | "MLS"       |
|     6 | "Musicals"  |
|     7 | "Plays"     |
|     8 | "Opera"     |
|     9 | "Pop"       |
|    10 | "Jazz"      |
|    11 | "Classical" |
+-------+-------------+
```

Funzione QUOTE_LITERAL

La funzione QUOTE_LITERAL restituisce la stringa specificata come una stringa con virgoletta singola in modo che possa essere utilizzata come letterale di stringa in un'istruzione SQL. Se il parametro di input è un numero, QUOTE_LITERAL lo considera come una stringa. Raddoppia in modo appropriato qualsiasi virgoletta singola incorporata e barra rovesciata.

Sintassi

```
QUOTE_LITERAL(string)
```

Argomento

stringa

Una stringa CHAR o VARCHAR.

Tipo restituito

La funzione QUOTE_LITERAL restituisce una stringa CHAR o VARCHAR che appartiene allo stesso tipo di dati della stringa di input.

Esempi

Per restituire la stringa `''CAT''` con virgolette SINGOLE, utilizza l'esempio seguente.

```
SELECT QUOTE_LITERAL('''CAT''');

+---------------+
| quote_literal |
+---------------+
| '''CAT'''     |
+---------------+
```

Negli esempi seguenti vengono utilizzati i dati della tabella CATEGORY database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire la colonna CATNAME racchiusa tra virgolette singole, utilizza l'esempio seguente.

```
SELECT catid, QUOTE_LITERAL(catname)
FROM category
ORDER BY 1,2;
```

```
+-------+---------------+
| catid | quote_literal |
+-------+---------------+
|     1 | 'MLB'         |
|     2 | 'NHL'         |
|     3 | 'NFL'         |
|     4 | 'NBA'         |
|     5 | 'MLS'         |
|     6 | 'Musicals'    |
|     7 | 'Plays'       |
|     8 | 'Opera'       |
|     9 | 'Pop'         |
|    10 | 'Jazz'        |
|    11 | 'Classical'   |
+-------+---------------+
```

Per restituire la colonna CATID racchiusa tra virgolette singole, utilizza l'esempio seguente.

```
SELECT QUOTE_LITERAL(catid), catname
FROM category
ORDER BY 1,2;

+---------------+-----------+
| quote_literal |  catname  |
+---------------+-----------+
| '1'           | MLB       |
| '10'          | Jazz      |
| '11'          | Classical |
| '2'           | NHL       |
| '3'           | NFL       |
| '4'           | NBA       |
| '5'           | MLS       |
| '6'           | Musicals  |
| '7'           | Plays     |
| '8'           | Opera     |
| '9'           | Pop       |
+---------------+-----------+
```

Funzione REGEXP_COUNT

Cerca una stringa per un modello di espressione regolare e restituisce un numero intero che indica il numero di volte in cui il modello si verifica nella stringa. Se non viene trovata alcuna corrispondenza,

la funzione restituisce 0. Per ulteriori informazioni sulle espressioni regolari, Operatori POSIX consultate Regular expression in Wikipedia.

Sintassi

```
REGEXP_COUNT( source_string, pattern [, position [, parameters ] ] )
```

Argomenti

source_string

Una stringa CHAR o VARCHAR.

pattern

Una stringa letterale UTF-8 che rappresenta un modello di espressione regolare. Per ulteriori informazioni, consulta Operatori POSIX.

posizione

(Facoltativo) Un valore INTEGER positivo che indica la posizione all'interno di source_string per iniziare la ricerca. La posizione si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. Il valore predefinito è 1. Se position è inferiore a 1, la ricerca inizia con il primo carattere di source_string. Se position è maggiore rispetto al numero di caratteri in source_string, il risultato è 0.

parameters

(Facoltativo) Uno o più letterali di stringa che indicano come la funzione corrisponde al modello. Di seguito sono riportati i valori possibili:

- c: eseguire una corrispondenza in base a maiuscole e minuscole. L'impostazione predefinita è utilizzare la corrispondenza con distinzione tra maiuscole e minuscole.
- i: eseguire una corrispondenza senza distinzione tra maiuscole e minuscole.
- p: interpreta il modello con il dialetto Perl Compatible Regular Expression (PCRE). Per ulteriori informazioni su PCRE, vedere Espressioni regolari compatibili con Perl in Wikipedia.

Tipo restituito

INTEGER

Esempi

Per contare il numero di volte in cui si verifica una sequenza di tre lettere, utilizza l'esempio seguente.

```
SELECT REGEXP_COUNT('abcdefghijklmnopqrstuvwxyz', '[a-z]{3}');

+--------------+
| regexp_count |
+--------------+
|            8 |
+--------------+
```

Per contare le occorrenze della stringa FOX utilizzando una corrispondenza senza distinzione tra maiuscole e minuscole, utilizza l'esempio seguente.

```
SELECT REGEXP_COUNT('the fox', 'FOX', 1, 'i');

+--------------+
| regexp_count |
+--------------+
|            1 |
+--------------+
```

Per utilizzare un modello scritto in dialetto PCRE per individuare le parole contenenti almeno un numero e una lettera minuscola, utilizza l'esempio seguente. L'esempio utilizza l'operatore ?=, che ha una connotazione look-ahead specifica in PCRE. In questo esempio viene contato il numero di ricorrenze di tali parole, con la corrispondenza tra maiuscole e minuscole.

```
SELECT REGEXP_COUNT('passwd7 plain A1234 a1234', '(?=[^ ]*[a-z])(?=[^ ]*[0-9])[^ ]+',
 1, 'p');

+--------------+
| regexp_count |
+--------------+
|            2 |
+--------------+
```

Per utilizzare un modello scritto in dialetto PCRE per individuare le parole contenenti almeno un numero e una lettera minuscola, utilizza l'esempio seguente. Utilizza l'operatore ?=, che ha una connotazione specifica in PCRE. In questo esempio viene contato il numero di ricorrenze di tali

parole, ma differisce dall'esempio precedente in quanto utilizza la corrispondenza senza distinzione tra maiuscole e minuscole.

```
SELECT REGEXP_COUNT('passwd7 plain A1234 a1234', '(?=[^ ]*[a-z])(?=[^ ]*[0-9])[^ ]+',
 1, 'ip');

+--------------+
| regexp_count |
+--------------+
|            3 |
+--------------+
```

Nell'esempio seguente vengono utilizzati i dati della tabella USERS database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per contare il numero di volte in cui il nome di dominio di livello superiore è `org` oppure `edu`, utilizza l'esempio seguente.

```
SELECT email, REGEXP_COUNT(email,'@[^.]*\.(org|edu)') FROM users
ORDER BY userid LIMIT 4;

+----------------------------------------------+--------------+
|                    email                     | regexp_count |
+----------------------------------------------+--------------+
| Etiam.laoreet.libero@sodalesMaurisblandit.edu |            1 |
| Suspendisse.tristique@nonnisiAenean.edu       |            1 |
| amet.faucibus.ut@condimentumegetvolutpat.ca   |            0 |
| sed@lacusUtnec.ca                             |            0 |
+----------------------------------------------+--------------+
```

Funzione REGEXP_INSTR

Cerca una stringa per un modello di espressione regolare e restituisce un integer che indica la posizione iniziale o finale della sottostringa corrispondente. Se non viene trovata alcuna corrispondenza, la funzione restituisce `0`. REGEXP_INSTR è simile alla funzione POSITION, ma consente di cercare una stringa per un modello di espressione regolare. Per ulteriori informazioni sulle espressioni regolari, vedere Operatori POSIX Espressione regolare in Wikipedia.

Sintassi

```
REGEXP_INSTR( source_string, pattern [, position [, occurrence] [, option [, parameters
 ] ] ] ] )
```

Argomenti

source_string

Un'espressione di stringa, come ad esempio un nome di colonna, da cercare.

pattern

Una stringa letterale UTF-8 che rappresenta un modello di espressione regolare. Per ulteriori informazioni, consulta Operatori POSIX.

posizione

(Facoltativo) Un valore `INTEGER` positivo che indica la posizione all'interno di source_string per iniziare la ricerca. La posizione si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. Il valore predefinito è 1. Se position è inferiore a 1, la ricerca inizia con il primo carattere di source_string. Se position è maggiore rispetto al numero di caratteri in source_string, il risultato è `0`.

occorrenza

(Facoltativo) Un valore `INTEGER` positivo che indica quale occorrenza del modello utilizzare. REGEXP_INSTR ignora le prime corrispondenze *`occurrence`*`-1`. Il valore predefinito è 1. Se occurrence è inferiore a 1 oppure maggiore rispetto al numero di caratteri in source_string, la ricerca viene ignorata e il risultato è `0`.

option

(Facoltativo) Un valore che indica se restituire la posizione del primo carattere della corrispondenza (`0`) o la posizione del primo carattere dopo la fine della corrispondenza (1). Un valore diverso da zero equivale a 1. Il valore predefinito è `0`.

parameters

(Facoltativo) Uno o più letterali di stringa che indicano come la funzione corrisponde al modello. Di seguito sono riportati i valori possibili:

- c: eseguire una corrispondenza in base a maiuscole e minuscole. L'impostazione predefinita è utilizzare la corrispondenza con distinzione tra maiuscole e minuscole.

- i: eseguire una corrispondenza senza distinzione tra maiuscole e minuscole.
- e: estrarre una sottostringa usando una sottoespressione.

  Se modello include una sottoespressione, REGEXP_INSTR corrisponde a una sottostringa che utilizza la prima sottoespressione in modello. REGEXP_INSTR considera solo la prima sottoespressione; le sottoespressioni aggiuntive vengono ignorate. Se il modello non ha una sottoespressione, REGEXP_INSTR ignora il parametro "e".
- p: interpreta il modello con il dialetto Perl Compatible Regular Expression (PCRE). Per ulteriori informazioni su PCRE, vedere Espressioni regolari compatibili con Perl in Wikipedia.

Tipo restituito

Numero intero

Esempi

Negli esempi seguenti vengono utilizzati i dati della tabella USERS database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per ricercare il carattere @ che inizia un nome di dominio e restituisce la posizione iniziale della prima corrispondenza, utilizza l'esempio seguente.

```
SELECT email, REGEXP_INSTR(email, '@[^.]*')
FROM users
ORDER BY userid LIMIT 4;

+----------------------------------------------+--------------+
|                    email                     | regexp_instr |
+----------------------------------------------+--------------+
| Etiam.laoreet.libero@sodalesMaurisblandit.edu |           21 |
| Suspendisse.tristique@nonnisiAenean.edu       |           22 |
| amet.faucibus.ut@condimentumegetvolutpat.ca   |           17 |
| sed@lacusUtnec.ca                             |            4 |
+----------------------------------------------+--------------+
```

Per ricercare le varianti della parola `Center` e restituire la posizione iniziale della prima corrispondenza, utilizza l'esempio seguente.

```
SELECT venuename, REGEXP_INSTR(venuename,'[cC]ent(er|re)$')
FROM venue
WHERE REGEXP_INSTR(venuename,'[cC]ent(er|re)$') > 0
```

```
ORDER BY venueid LIMIT 4;

+----------------------+--------------+
|       venuename      | regexp_instr |
+----------------------+--------------+
| The Home Depot Center |           16 |
| Izod Center           |            6 |
| Wachovia Center       |           10 |
| Air Canada Centre     |           12 |
+----------------------+--------------+
```

Per trovare la posizione iniziale della prima occorrenza della stringa `FOX` utilizzando una logica di associazione senza distinzione tra maiuscole e minuscole, utilizza l'esempio seguente.

```
SELECT REGEXP_INSTR('the fox', 'FOX', 1, 1, 0, 'i');

+--------------+
| regexp_instr |
+--------------+
|            5 |
+--------------+
```

Per utilizzare un modello scritto in dialetto PCRE per individuare le parole contenenti almeno un numero e una lettera minuscola, utilizza l'esempio seguente. Utilizza l'operatore `?=`, che ha una connotazione look-ahead specifica in PCRE. In questo esempio viene trovata la posizione iniziale della seconda parola di questo tipo.

```
SELECT REGEXP_INSTR('passwd7 plain A1234 a1234', '(?=[^ ]*[a-z])(?=[^ ]*[0-9])[^ ]+',
 1, 2, 0, 'p');

+--------------+
| regexp_instr |
+--------------+
|           21 |
+--------------+
```

Per utilizzare un modello scritto in dialetto PCRE per individuare le parole contenenti almeno un numero e una lettera minuscola, utilizza l'esempio seguente. Utilizza l'operatore `?=`, che ha una connotazione look-ahead specifica in PCRE. In questo esempio viene trovata la posizione iniziale della seconda parola, ma differisce dall'esempio precedente in quanto utilizza la corrispondenza senza distinzione tra maiuscole e minuscole.

```
SELECT REGEXP_INSTR('passwd7 plain A1234 a1234', '(?=[^ ]*[a-z])(?=[^ ]*[0-9])[^ ]+',
 1, 2, 0, 'ip');

+--------------+
| regexp_instr |
+--------------+
|           15 |
+--------------+
```

Funzione REGEXP_REPLACE

Cerca una stringa per un modello di espressione regolare e sostituisce ogni occorrenza del modello con la stringa specificata. REGEXP_REPLACE è simile a Funzione REPLACE, ma consente di cercare una stringa per un modello di espressione regolare. Per ulteriori informazioni sulle espressioni regolari, vedere Operatori POSIX Espressione regolare in Wikipedia.

REGEXP_REPLACE è simile a Funzione TRANSLATE e a Funzione REPLACE, ad eccezione del fatto che TRANSLATE esegue più sostituzioni a carattere singolo e REPLACE sostituisce un'intera stringa con un'altra stringa, mentre REGEXP_REPLACE consente di cercare una stringa per un modello di espressione regolare.

Sintassi

```
REGEXP_REPLACE( source_string, pattern [, replace_string [ , position [, parameters
 ] ] ] )
```

Argomenti

source_string

Un'espressione di stringa CHAR o VARCHAR, come ad esempio un nome di colonna, da cercare.

pattern

Una stringa letterale UTF-8 che rappresenta un modello di espressione regolare. Per ulteriori informazioni, consulta Operatori POSIX.

replace_string

(Facoltativo) Un'espressione di stringa CHAR or VARCHAR, ad esempio un nome di colonna, che sostituirà ogni occorrenza del modello. L'impostazione predefinita è una stringa vuota ("").

posizione

(Facoltativo) Un numero intero positivo che indica la posizione all'interno di source_string per iniziare la ricerca. La posizione si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. Il valore predefinito è 1. Se position è inferiore a 1, la ricerca inizia con il primo carattere di source_string. Se posizione è maggiore rispetto al numero di caratteri in source_string, il risultato è source_string.

parameters

(Facoltativo) Uno o più letterali di stringa che indicano come la funzione corrisponde al modello. Di seguito sono riportati i valori possibili:

- c: eseguire una corrispondenza in base a maiuscole e minuscole. L'impostazione predefinita è utilizzare la corrispondenza con distinzione tra maiuscole e minuscole.
- i: eseguire una corrispondenza senza distinzione tra maiuscole e minuscole.
- p: interpreta il modello con il dialetto Perl Compatible Regular Expression (PCRE). Per ulteriori informazioni su PCRE, vedere Espressioni regolari compatibili con Perl in Wikipedia.

Tipo restituito

VARCHAR

Se pattern o replace_string è `NULL`, la funzione restituisce `NULL`.

Esempi

Per sostituire tutte le ricorrenze della stringa `FOX` nel valore `quick brown fox` usando una corrispondenza senza distinzione tra maiuscole e minuscole, utilizza l'esempio seguente.

```
SELECT REGEXP_REPLACE('the fox', 'FOX', 'quick brown fox', 1, 'i');

+----------------------+
|   regexp_replace     |
+----------------------+
| the quick brown fox |
+----------------------+
```

Nell'esempio seguente viene utilizzato un modello scritto in dialetto PCRE per individuare le parole contenenti almeno un numero e una lettera minuscola. Utilizza l'operatore ?=, che ha una

connotazione look-ahead specifica in PCRE. Per sostituire ogni occorrenza di tale parola con il valore `[hidden]`, utilizza l'esempio seguente.

```
SELECT REGEXP_REPLACE('passwd7 plain A1234 a1234', '(?=[^ ]*[a-z])(?=[^ ]*[0-9])[^ ]+',
 '[hidden]', 1, 'p');

+-------------------------------+
|        regexp_replace         |
+-------------------------------+
| [hidden] plain A1234 [hidden] |
+-------------------------------+
```

Nell'esempio seguente viene utilizzato un modello scritto in dialetto PCRE per individuare le parole contenenti almeno un numero e una lettera minuscola. Utilizza l'operatore ?=, che ha una connotazione look-ahead specifica in PCRE. Per sostituire ogni ricorrenza di tale parola con il valore `[hidden]`, ma diversamente dall'esempio precedente in quanto utilizza la corrispondenza senza distinzione tra maiuscole e minuscole, utilizza l'esempio seguente.

```
SELECT REGEXP_REPLACE('passwd7 plain A1234 a1234', '(?=[^ ]*[a-z])(?=[^ ]*[0-9])[^ ]+',
 '[hidden]', 1, 'ip');

+---------------------------------+
|          regexp_replace         |
+---------------------------------+
| [hidden] plain [hidden] [hidden] |
+---------------------------------+
```

Negli esempi seguenti vengono utilizzati i dati della tabella USERS database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per eliminare @ e il nome di dominio dagli indirizzi e-mail, utilizza l'esempio seguente.

```
SELECT email, REGEXP_REPLACE(email, '@.*\\.(org|gov|com|edu|ca)$')
FROM users
ORDER BY userid LIMIT 4;

+-----------------------------------------------+----------------------+
|                     email                     |    regexp_replace    |
+-----------------------------------------------+----------------------+
| Etiam.laoreet.libero@sodalesMaurisblandit.edu | Etiam.laoreet.libero |
| Suspendisse.tristique@nonnisiAenean.edu       | Suspendisse.tristique |
```

```
| amet.faucibus.ut@condimentumegetvolutpat.ca   | amet.faucibus.ut      |
| sed@lacusUtnec.ca                             | sed                   |
+-----------------------------------------------+-----------------------+
```

Per sostituire i nomi di dominio degli indirizzi e-mail con `internal.company.com`, utilizza l'esempio seguente.

```
SELECT email, REGEXP_REPLACE(email, '@.*\\.[[:alpha:]]{2,3}','@internal.company.com')
FROM users
ORDER BY userid LIMIT 4;

+-----------------------------------------------
+-------------------------------------------+
|                     email                     |               regexp_replace
      |
+-----------------------------------------------
+-------------------------------------------+
| Etiam.laoreet.libero@sodalesMaurisblandit.edu |
 Etiam.laoreet.libero@internal.company.com  |
| Suspendisse.tristique@nonnisiAenean.edu       |
 Suspendisse.tristique@internal.company.com |
| amet.faucibus.ut@condimentumegetvolutpat.ca   | amet.faucibus.ut@internal.company.com
      |
| sed@lacusUtnec.ca                             | sed@internal.company.com
      |
+-----------------------------------------------
+-------------------------------------------+
```

Funzione REGEXP_SUBSTR

Restituisce i caratteri da una stringa cercando un modello di espressione regolare. REGEXP_SUBSTR è simile alla funzione Funzione SUBSTRING ma consente di cercare una stringa per un modello di espressione regolare. Se la funzione non riesce a far corrispondere l'espressione regolare ad alcun carattere della stringa, restituisce una stringa vuota. Per ulteriori informazioni sulle espressioni regolari, vedere Operatori POSIX Espressione regolare in Wikipedia.

Sintassi

```
REGEXP_SUBSTR( source_string, pattern [, position [, occurrence [, parameters ] ] ] )
```

Argomenti

source_string

Un'espressione della stringa da ricercare.

pattern

Una stringa letterale UTF-8 che rappresenta un modello di espressione regolare. Per ulteriori informazioni, consulta Operatori POSIX.

posizione

Un integer positivo che indica la posizione all'interno di source_string per iniziare la ricerca. La posizione si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. Il valore di default è 1. Se posizione è inferiore a 1, la ricerca inizia con il primo carattere di source_string. Se posizione è maggiore rispetto al numero di caratteri in source_string, il risultato è una stringa vuota ("").

occorrenza

Un integer positivo che indica quale occorrenza del modello utilizzare. REGEXP_SUBSTR salta le prime corrispondenze occorrenza -1. Il valore di default è 1. Se occorrenza è inferiore a 1 oppure maggiore rispetto al numero di caratteri in source_string, la ricerca viene ignorata e il risultato è NULL.

parameters

Uno o più letterali di stringa che indicano come la funzione corrisponde al modello. Di seguito sono riportati i valori possibili:

- c: eseguire una corrispondenza in base a maiuscole e minuscole. L'impostazione predefinita è utilizzare la corrispondenza con distinzione tra maiuscole e minuscole.
- i: eseguire una corrispondenza senza distinzione tra maiuscole e minuscole.
- e: estrarre una sottostringa usando una sottoespressione.

  Se modello include una sottoespressione, REGEXP_SUBSTR corrisponde a una sottostringa che utilizza la prima sottoespressione in modello. Un'espressione secondaria è un'espressione all'interno del modello racchiusa tra parentesi. Ad esempio, per il modello `'This is a (\\w+)'` corrisponde alla prima espressione con la stringa `'This is a '` seguita da una parola. Invece di restituire un modello, REGEXP_SUBSTR con il parametro e restituisce solo la stringa all'interno dell'espressione secondaria.

REGEXP_SUBSTR considera solo la prima sottoespressione; le sottoespressioni aggiuntive vengono ignorate. Se il modello non ha una sottoespressione, REGEXP_SUBSTR ignora il parametro "e".

- p: interpreta il modello con il dialetto Perl Compatible Regular Expression (PCRE). Per ulteriori informazioni su PCRE, vedere Espressioni regolari compatibili con Perl in Wikipedia.

Tipo restituito

VARCHAR

Esempi

L'esempio seguente restituisce la porzione di un indirizzo email tra il carattere @ e l'estensione del dominio. I dati `users` richiesti provengono dai dati di esempio di Amazon Redshift. Per ulteriori informazioni, consulta Database di esempio.

```
SELECT email, regexp_substr(email,'@[^.]*')
FROM users
ORDER BY userid LIMIT 4;

                     email                     |      regexp_substr
-----------------------------------------------+--------------------------
 Suspendisse.tristique@nonnisiAenean.edu       | @nonnisiAenean
 amet.faucibus.ut@condimentumegetvolutpat.ca   | @condimentumegetvolutpat
 sed@lacusUtnec.ca                             | @lacusUtnec
 Cum@accumsan.com                              | @accumsan
```

Nell'esempio seguente viene restituita la porzione dell'input corrispondente alla prima ricorrenza della stringa `FOX` utilizzando una corrispondenza senza distinzione tra maiuscole e minuscole.

```
SELECT regexp_substr('the fox', 'FOX', 1, 1, 'i');

 regexp_substr
---------------
 fox
```

Nell'esempio seguente viene restituita la porzione dell'input corrispondente alla seconda occorrenza della stringa `FOX` utilizzando una corrispondenza senza distinzione tra maiuscole e minuscole. Il risultato è `NULL` (vuoto) perché non esiste una seconda occorrenza.

```
SELECT regexp_substr('the fox', 'FOX', 1, 2, 'i');

 regexp_substr
---------------
```

L'istruzione di esempio seguente restituisce la prima parte dell'input che inizia con lettere minuscole. Dal punto di vista funzionale è identica alla medesima istruzione SELECT senza il parametro `c`.

```
SELECT regexp_substr('THE SECRET CODE IS THE LOWERCASE PART OF 1931abc0EZ.', '[a-z]+',
 1, 1, 'c');

 regexp_substr
---------------
 abc
```

Nell'esempio seguente viene utilizzato un modello scritto in dialetto PCRE per individuare le parole contenenti almeno un numero e una lettera minuscola. Utilizza l'operatore ?=, che ha una connotazione look-ahead specifica in PCRE. In questo esempio viene restituita la parte dell'input corrispondente alla seconda parola di questo tipo.

```
SELECT regexp_substr('passwd7 plain A1234 a1234', '(?=[^ ]*[a-z])(?=[^ ]*[0-9])[^ ]+',
 1, 2, 'p');

 regexp_substr
---------------
 a1234
```

Nell'esempio seguente viene utilizzato un modello scritto in dialetto PCRE per individuare le parole contenenti almeno un numero e una lettera minuscola. Utilizza l'operatore ?=, che ha una connotazione look-ahead specifica in PCRE. In questo esempio viene restituita la parte di input corrispondente alla seconda parola, ma differisce dall'esempio precedente in quanto si utilizza la corrispondenza senza distinzione tra maiuscole e minuscole.

```
SELECT regexp_substr('passwd7 plain A1234 a1234', '(?=[^ ]*[a-z])(?=[^ ]*[0-9])[^ ]+',
 1, 2, 'ip');

 regexp_substr
---------------
 A1234
```

L'esempio seguente utilizza un'espressione secondaria per trovare la seconda stringa che corrisponde al modello `'this is a (\\w+)'` utilizzando la corrispondenza senza distinzione tra maiuscole e minuscole. Restituisce l'espressione secondaria tra parentesi.

```
SELECT regexp_substr(
               'This is a cat, this is a dog. This is a mouse.',
               'this is a (\\w+)', 1, 2, 'ie');

 regexp_substr
---------------
 dog
```

Funzione REPEAT

Ripete una stringa il numero specificato di volte. Se il parametro di input è numerico, REPEAT lo considera come una stringa.

Sinonimo di Funzione REPLICATE.

Sintassi

```
REPEAT(string, integer)
```

Argomenti

stringa

Il primo parametro di input è la stringa da ripetere.

integer

Il secondo parametro è un valore `INTEGER` che indica il numero di volte in cui ripetere la stringa.

Tipo restituito

VARCHAR

Esempi

Nell'esempio seguente vengono utilizzati i dati della tabella CATEGORY database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per ripetere il valore della colonna CATID nella tabella CATEGORY tre volte, utilizza l'esempio seguente.

```
SELECT catid, REPEAT(catid,3)
FROM category
ORDER BY 1,2;

+-------+--------+
| catid | repeat |
+-------+--------+
|     1 |    111 |
|     2 |    222 |
|     3 |    333 |
|     4 |    444 |
|     5 |    555 |
|     6 |    666 |
|     7 |    777 |
|     8 |    888 |
|     9 |    999 |
|    10 | 101010 |
|    11 | 111111 |
+-------+--------+
```

Funzione REPLACE

Sostituisce tutte le occorrenze di un insieme di caratteri all'interno di una stringa esistente con altri caratteri specificati.

REPLACE è simile a Funzione TRANSLATE e a Funzione REGEXP_REPLACE, ad eccezione del fatto che TRANSLATE esegue più sostituzioni a carattere singolo e REGEXP_REPLACE consente di cercare una stringa per un modello di espressione regolare, mentre REPLACE sostituisce un'intera stringa con un'altra stringa.

Sintassi

```
REPLACE(string, old_chars, new_chars)
```

Argomenti

stringa

La stringa CHAR o VARCHAR da cercare in ricerca

old_chars

La stringa CHAR o VARCHAR da sostituire.

new_chars

Nuova stringa CHAR o VARCHAR che sostituisce old_string.

Tipo restituito

VARCHAR

Se old_chars o new_chars è NULL, il risultato è NULL.

Esempi

Nell'esempio seguente vengono utilizzati i dati della tabella CATEGORY database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per convertire la stringa Shows in Theatre nel campo CATGROUP, utilizza l'esempio seguente.

```
SELECT catid, catgroup, REPLACE(catgroup, 'Shows', 'Theatre')
FROM category
ORDER BY 1,2,3;

+-------+----------+----------+
| catid | catgroup | replace  |
+-------+----------+----------+
|     1 | Sports   | Sports   |
|     2 | Sports   | Sports   |
|     3 | Sports   | Sports   |
|     4 | Sports   | Sports   |
|     5 | Sports   | Sports   |
|     6 | Shows    | Theatre  |
|     7 | Shows    | Theatre  |
|     8 | Shows    | Theatre  |
|     9 | Concerts | Concerts |
|    10 | Concerts | Concerts |
|    11 | Concerts | Concerts |
+-------+----------+----------+
```

Funzione REPLICATE

Sinonimo della funzione REPEAT.

Per informazioni, consultare Funzione REPEAT.

Funzione REVERSE

La funzione REVERSE funziona su una stringa e restituisce i caratteri in ordine inverso. Ad esempio, `reverse('abcde')` restituisce `edcba`. Questa funzione funziona su tipi di dati numerici e di date, così come su tipi di dati di carattere; tuttavia, nella maggior parte dei casi ha un valore pratico per le stringhe di caratteri.

Sintassi

```
REVERSE( expression )
```

Argomento

espressione

Un'espressione con un carattere, una data, un timestamp o un tipo di dati numerici che rappresenta la destinazione dell'inversione di caratteri. Tutte le espressioni vengono convertite implicitamente in stringhe `VARCHAR`. Spazi vuoti finali nelle stringhe `CHAR` vengono ignorati.

Tipo restituito

VARCHAR

Esempi

Gli esempi seguenti utilizzano i dati delle tabelle USERS e SALES database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per selezionare cinque nomi di città distinti e i corrispondenti nomi invertiti dalla tabella USERS, utilizza l'esempio seguente.

```
SELECT DISTINCT city AS cityname, REVERSE(cityname)
FROM users
ORDER BY city LIMIT 5;
```

```
+----------+----------+
| cityname | reverse  |
+----------+----------+
| Aberdeen | needrebA |
| Abilene  | enelibA  |
| Ada      | adA      |
| Agat     | tagA     |
| Agawam   | mawagA   |
+----------+----------+
```

Per selezionare cinque ID di vendita e i relativi ID invertiti corrispondenti convertiti come stringhe di caratteri, utilizza l'esempio seguente.

```
SELECT salesid, REVERSE(salesid)
FROM sales
ORDER BY salesid DESC LIMIT 5;

+---------+---------+
| salesid | reverse |
+---------+---------+
|  172456 |  654271 |
|  172455 |  554271 |
|  172454 |  454271 |
|  172453 |  354271 |
|  172452 |  254271 |
+---------+---------+
```

Funzione RTRIM

La funzione RTRIM riduce un insieme specificato di caratteri dalla fine di una stringa. Rimuove la stringa più lunga contenente solo i caratteri nell'elenco dei caratteri di taglio. Il taglio è completo quando un carattere di taglio non appare nella stringa di input.

Sintassi

```
RTRIM( string, trim_chars )
```

Argomenti

stringa

Una stringa, una colonna, un'espressione o una stringa letterale da tagliare.

trim_chars

Una colonna o un'espressione di stringa o un valore letterale di stringa che rappresenta i caratteri da tagliare dall'inizio della stringa. Se non specificato, viene utilizzato uno spazio come carattere di taglio.

Tipo restituito

Una stringa che è lo stesso tipo di dati dell'argomento stringa.

Esempio

L'esempio seguente riduce gli spazi vuoti iniziali e finali dalla stringa `' abc '`:

```
select '     abc    ' as untrim, rtrim('     abc    ') as trim;

untrim    | trim
----------+------
   abc    |    abc
```

L'esempio seguente rimuove le stringhe `'xyz'` iniziali dalla stringa `'xyzaxyzbxyzcxyz'`. Le occorrenze finali di `'xyz'` vengono rimosse, ma le occorrenze interne alla stringa non vengono rimosse.

```
select 'xyzaxyzbxyzcxyz' as untrim,
rtrim('xyzaxyzbxyzcxyz', 'xyz') as trim;

     untrim      |   trim
-----------------+-----------
 xyzaxyzbxyzcxyz | xyzaxyzbxyzc
```

L'esempio seguente rimuove le parti finali dalla stringa `'setuphistorycassettes'` che corrispondono a uno qualsiasi dei caratteri nell'elenco `'tes'` trim_chars. Qualsiasi carattere `t`, `e` o `s` che si verifica prima di un altro carattere che non è nell'elenco trim_chars alla fine della stringa di input viene rimosso.

```
SELECT rtrim('setuphistorycassettes', 'tes');

     rtrim
-----------------
```

```
 setuphistoryca
```

L'esempio seguente riduce i caratteri "Parco"dalla fine di VENUENAME laddove presente:

```
select venueid, venuename, rtrim(venuename, 'Park')
from venue
order by 1, 2, 3
limit 10;

venueid |         venuename          |          rtrim
--------+----------------------------+-------------------------
      1 | Toyota Park                | Toyota
      2 | Columbus Crew Stadium      | Columbus Crew Stadium
      3 | RFK Stadium                | RFK Stadium
      4 | CommunityAmerica Ballpark  | CommunityAmerica Ballp
      5 | Gillette Stadium           | Gillette Stadium
      6 | New York Giants Stadium    | New York Giants Stadium
      7 | BMO Field                  | BMO Field
      8 | The Home Depot Center      | The Home Depot Cente
      9 | Dick's Sporting Goods Park | Dick's Sporting Goods
     10 | Pizza Hut Park             | Pizza Hut
```

Si noti che RTRIM rimuove tutti i caratteri in `P`, `a` , `r` oppure `k` quando appaiono alla fine di un VENUENAME.

Funzione SOUNDEX

La funzione SOUNDEX restituisce il valore American Soundex costituito dalla prima lettera della stringa di input seguita da una codifica a 3 cifre dei suoni che rappresentano la pronuncia inglese della stringa specificata. Ad esempio `Smith` e `Smyth` hanno lo stesso valore Soundex.

Sintassi

```
SOUNDEX(string)
```

Argomenti

stringa

Specifica una stringa `CHAR` o `VARCHAR` che desideri convertire in un valore di codice American Soundex.

Tipo restituito

VARCHAR(4)

Note per l'utilizzo

La funzione SOUNDEX converte solo caratteri ASCII alfabetici minuscoli o maiuscoli inglesi, inclusi a-z e A-Z. SOUNDEX ignora gli altri caratteri. SOUNDEX restituisce un singolo valore Soundex per una stringa di più parole separate da spazi.

```
SELECT SOUNDEX('AWS Amazon');

+---------+
| soundex |
+---------+
| A252    |
+---------+
```

SOUNDEX restituisce una stringa vuota se la stringa di input non contiene lettere inglesi.

```
SELECT SOUNDEX('+-*/%');

+---------+
| soundex |
+---------+
|         |
+---------+
```

Esempi

Per restituire il valore Soundex per `Amazon`, utilizza l'esempio seguente.

```
SELECT SOUNDEX('Amazon');

+---------+
| soundex |
+---------+
| A525    |
+---------+
```

Per restituire il valore Soundex per `smith` e `smyth`, utilizza l'esempio seguente. Tieni presente che i valori di Soundex sono gli stessi.

```
SELECT SOUNDEX('smith'), SOUNDEX('smyth');

+-------+-------+
| smith | smyth |
+-------+-------+
| S530  | S530  |
+-------+-------+
```

Funzione SPLIT_PART

Divide una stringa sul delimitatore specificato e restituisce la parte nella posizione specificata.

Sintassi

```
SPLIT_PART(string, delimiter, position)
```

Argomenti

stringa

Una stringa, una colonna, un'espressione o una stringa letterale da dividere. La stringa può essere CHAR o VARCHAR.

delimiter

La stringa del delimitatore che indica le sezioni della stringa di input.

Se delimiter è un letterale, racchiuderlo tra virgolette singole.

posizione

Posizione della porzione di stringa da restituire (contando da 1). Deve essere un integer superiore a 0. Se posizione è maggiore del numero di porzioni di stringa, SPLIT_PART restituisce una stringa vuota. Se il delimitatore non si trova nella stringa, il valore restituito contiene i contenuti della parte specificata, che possono essere l'intera stringa o un valore vuoto.

Tipo restituito

Una stringa CHAR o VARCHAR, uguale al parametro di stringa.

Esempi

L'esempio seguente divide una stringa letterale in parti utilizzando il delimitatore $ e restituisce la seconda parte.

```
select split_part('abc$def$ghi','$',2)

split_part
----------
def
```

L'esempio seguente divide una stringa letterale in parti utilizzando il delimitatore $. Restituisce una stringa vuota perché la parte 4 non viene trovata.

```
select split_part('abc$def$ghi','$',4)

split_part
----------

```

L'esempio seguente divide una stringa letterale in parti utilizzando il delimitatore #. Restituisce l'intera stringa, che è la prima parte, perché il delimitatore non è stato trovato.

```
select split_part('abc$def$ghi','#',1)

split_part
------------
abc$def$ghi
```

L'esempio seguente divide il campo timestamp LISTTIME in componenti anno, mese e data.

```
select listtime, split_part(listtime,'-',1) as year,
split_part(listtime,'-',2) as month,
split_part(split_part(listtime,'-',3),' ',1) as day
from listing limit 5;

      listtime       | year | month | day
---------------------+------+-------+------
 2008-03-05 12:25:29 | 2008 | 03    | 05
 2008-09-09 08:03:36 | 2008 | 09    | 09
 2008-09-26 05:43:12 | 2008 | 09    | 26
```

```
 2008-10-04 02:00:30 | 2008 | 10    | 04
 2008-01-06 08:33:11 | 2008 | 01    | 06
```

L'esempio seguente seleziona il campo timestamp LISTTIME e lo divide sul carattere `'-'` per ottenere il mese (la seconda parte della stringa LISTTIME), quindi conta il numero di voci per ogni mese:

```
select split_part(listtime,'-',2) as month, count(*)
from listing
group by split_part(listtime,'-',2)
order by 1, 2;

 month | count
-------+-------
    01 | 18543
    02 | 16620
    03 | 17594
    04 | 16822
    05 | 17618
    06 | 17158
    07 | 17626
    08 | 17881
    09 | 17378
    10 | 17756
    11 | 12912
    12 | 4589
```

Funzione STRPOS

Restituisce la posizione di una sottostringa specificata all'interno di una stringa specificata.

Per funzioni simili, consulta Funzione CHARINDEX e Funzione POSITION.

Sintassi

```
STRPOS(string, substring )
```

Argomenti

stringa

Il primo parametro di input è la stringa CHAR o VARCHAR da cercare.

sottostringa

Il secondo parametro è la sottostringa da cercare all'interno della stringa.

Tipo restituito

INTEGER

La funzione STRPOS restituisce un valore `INTEGER` corrispondente alla posizione della sottostringa (basata su uno, non su zero). La posizione si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli.

Note per l'utilizzo

STRPOS restituisce `0` se la sottostringa non si trova all'interno della stringa.

```
SELECT STRPOS('dogfish', 'fist');

+--------+
| strpos |
+--------+
|      0 |
+--------+
```

Esempi

Per mostrare la posizione di `fish` in `dogfish`, utilizza l'esempio seguente.

```
SELECT STRPOS('dogfish', 'fish');

+--------+
| strpos |
+--------+
|      4 |
+--------+
```

Nell'esempio seguente vengono utilizzati i dati della tabella SALES del database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per restituire il numero di transazioni di vendita con COMMISSION superiore a 999,00 dalla tabella SALES, utilizza l'esempio seguente.

```
SELECT DISTINCT STRPOS(commission, '.'),
COUNT (STRPOS(commission, '.'))
FROM sales
WHERE STRPOS(commission, '.') > 4
GROUP BY STRPOS(commission, '.')
ORDER BY 1, 2;

+--------+-------+
| strpos | count |
+--------+-------+
|      5 |   629 |
+--------+-------+
```

Funzione STRTOL

Converte un'espressione di stringa di un numero della base specificata nel valore intero equivalente. Il valore convertito deve essere compreso nell'intervallo 64-bit firmato.

Sintassi

```
STRTOL(num_string, base)
```

Argomenti

num_string

Espressione di stringa di un numero da convertire. Se num_string è vuoto (`''`) o inizia con il carattere null (`'\0'`), il valore convertito è `0`. Se num_string è una colonna contenente un valore NULL, STRTOL restituisce `NULL`. La stringa può iniziare con qualsiasi quantità di spazio bianco, opzionalmente seguita da un singolo segno più '+' o meno '-' per indicare il positivo o il negativo. Il valore di default è '+'. Se base è 16, la stringa può opzionalmente iniziare con '`0x`'.

base

`INTEGER` tra 2 e 36.

Tipo restituito

BIGINT

Se num_string è null, la funzione restituisce `NULL`.

Esempi

Per convertire le coppie di valori di stringa e di base in numeri interi, utilizza l'esempio seguente.

```
SELECT STRTOL('0xf',16);

+--------+
| strtol |
+--------+
|     15 |
+--------+

SELECT STRTOL('abcd1234',16);

+------------+
|   strtol   |
+------------+
| 2882343476 |
+------------+

SELECT STRTOL('1234567', 10);

+---------+
| strtol  |
+---------+
| 1234567 |
+---------+

SELECT STRTOL('1234567', 8);

+--------+
| strtol |
+--------+
| 342391 |
+--------+

SELECT STRTOL('110101', 2);

+--------+
| strtol |
+--------+
|     53 |
+--------+
```

```
SELECT STRTOL('\0', 2);

+--------+
| strtol |
+--------+
|      0 |
+--------+
```

Funzione SUBSTRING

Restituisce il sottoinsieme di una stringa basata su una posizione iniziale specificata.

Se l'input è una stringa di carattere, la posizione iniziale e il numero di caratteri estratti si basano sui caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. Se l'input è un'espressione binaria, la posizione iniziale e la sottostringa estratta sono basate su byte. Non è possibile specificare una lunghezza negativa, ma è possibile specificare una posizione di partenza negativa.

Sintassi

```
SUBSTRING(character_string FROM start_position [ FOR number_characters ] )
```

```
SUBSTRING(character_string, start_position, number_characters )
```

```
SUBSTRING(binary_expression, start_byte, number_bytes )
```

```
SUBSTRING(binary_expression, start_byte )
```

Argomenti

character_string

La stringa da cercare. I tipi di dati non carattere sono trattati come una stringa.

start_position

La posizione all'interno della stringa per iniziare l'estrazione, a partire da 1. La start_position si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. Questo numero può essere negativo.

number_characters

Il numero di caratteri da estrarre (la lunghezza della sottostringa). Il number_characters si basa sul numero di caratteri, non di byte, pertanto i caratteri multibyte vengono contati come caratteri singoli. Questo numero non può essere negativo.

binary_expression

L'espressione binaria del tipo di dati VARBYTE da cercare.

start_byte

La posizione all'interno dell'espressione binaria per iniziare l'estrazione, a partire da 1. Questo numero può essere negativo.

number_byte

Il numero di byte da estrarre, ovvero, la lunghezza della sottostringa. Questo numero non può essere negativo.

Tipo restituito

VARCHAR o VARBYTE a seconda dell'input

Note per l'utilizzo

Di seguito sono riportati alcuni esempi di come è possibile utilizzare start_position e number_characters per estrarre sottostringhe da varie posizioni in una stringa.

L'esempio seguente restituisce una stringa di quattro caratteri che inizia con il sesto carattere.

```
select substring('caterpillar',6,4);
substring
-----------
pill
(1 row)
```

Se posizione_iniziale + numero_caratteri supera la lunghezza della stringa, SUBSTRING restituisce una sottostringa che inizia dalla posizione_iniziale fino alla fine della stringa. Ad esempio:

```
select substring('caterpillar',6,8);
```

```
substring
-----------
pillar
(1 row)
```

Se la `start_position` è negativa o pari a 0, la funzione SUBSTRING restituisce una sottostringa che inizia dal primo carattere di stringa con una lunghezza di `start_position` + `number_characters` -1. Ad esempio:

```
select substring('caterpillar',-2,6);
substring
-----------
cat
(1 row)
```

Se `start_position` + `number_characters` -1 è inferiore o pari a zero, SUBSTRING restituisce una stringa vuota. Ad esempio:

```
select substring('caterpillar',-5,4);
substring
-----------

(1 row)
```

Esempi

L'esempio seguente restituisce il mese dalla stringa LISTTIME nella tabella LISTING:

```
select listid, listtime,
substring(listtime, 6, 2) as month
from listing
order by 1, 2, 3
limit 10;

 listid |      listtime       | month
--------+---------------------+-------
      1 | 2008-01-24 06:43:29 | 01
      2 | 2008-03-05 12:25:29 | 03
      3 | 2008-11-01 07:35:33 | 11
      4 | 2008-05-24 01:18:37 | 05
      5 | 2008-05-17 02:29:11 | 05
```

```
      6 | 2008-08-15 02:08:13 | 08
      7 | 2008-11-15 09:38:15 | 11
      8 | 2008-11-09 05:07:30 | 11
      9 | 2008-09-09 08:03:36 | 09
     10 | 2008-06-17 09:44:54 | 06
(10 rows)
```

L'esempio seguente è lo stesso di sopra, ma utilizza l'opzione FROM...FOR:

```
select listid, listtime,
substring(listtime from 6 for 2) as month
from listing
order by 1, 2, 3
limit 10;

 listid |      listtime       | month
--------+---------------------+-------
      1 | 2008-01-24 06:43:29 | 01
      2 | 2008-03-05 12:25:29 | 03
      3 | 2008-11-01 07:35:33 | 11
      4 | 2008-05-24 01:18:37 | 05
      5 | 2008-05-17 02:29:11 | 05
      6 | 2008-08-15 02:08:13 | 08
      7 | 2008-11-15 09:38:15 | 11
      8 | 2008-11-09 05:07:30 | 11
      9 | 2008-09-09 08:03:36 | 09
     10 | 2008-06-17 09:44:54 | 06
(10 rows)
```

Non è possibile utilizzare SUBSTRING per estrarre in modo prevedibile il prefisso di una stringa che potrebbe contenere caratteri multibyte poiché è necessario specificare la lunghezza di una stringa multibyte in base al numero di byte, non al numero di caratteri. Per estrarre il segmento iniziale di una stringa in base alla lunghezza in byte, è possibile eseguire il CAST della stringa come VARCHAR(byte_length) per troncare la stringa, laddove byte_length è la lunghezza necessaria. L'esempio seguente estrae i primi 5 byte dalla stringa `'Fourscore and seven'`.

```
select cast('Fourscore and seven' as varchar(5));

varchar
-------
Fours
```

L'esempio seguente mostra una posizione iniziale negativa di un valore binario `abc`. Poiché la posizione iniziale è -3, la sottostringa viene estratta dall'inizio del valore binario. Il risultato viene visualizzato automaticamente come rappresentazione esadecimale della sottostringa binaria.

```
select substring('abc'::varbyte, -3);

 substring
-----------
 616263
```

L'esempio seguente mostra un valore 1 per la posizione iniziale di un valore binario `abc`. Poiché non è specificata la lunghezza, la stringa viene estratta dalla posizione iniziale alla fine della stringa. Il risultato viene visualizzato automaticamente come rappresentazione esadecimale della sottostringa binaria.

```
select substring('abc'::varbyte, 1);

 substring
-----------
 616263
```

L'esempio seguente mostra un 3 per la posizione iniziale di un valore binario `abc`. Poiché non è specificata la lunghezza, la stringa viene estratta dalla posizione iniziale alla fine della stringa. Il risultato viene visualizzato automaticamente come rappresentazione esadecimale della sottostringa binaria.

```
select substring('abc'::varbyte, 3);

 substring
-----------
 63
```

L'esempio seguente mostra un 2 per la posizione iniziale di un valore binario `abc`. La stringa viene estratta dalla posizione iniziale alla posizione 10, ma la fine della stringa si trova nella posizione 3. Il risultato viene visualizzato automaticamente come rappresentazione esadecimale della sottostringa binaria.

```
select substring('abc'::varbyte, 2, 10);
```

```
 substring
-----------
 6263
```

L'esempio seguente mostra un 2 per la posizione iniziale di un valore binario `abc`. La stringa viene estratta dalla posizione iniziale per 1 byte. Il risultato viene visualizzato automaticamente come rappresentazione esadecimale della sottostringa binaria.

```
select substring('abc'::varbyte, 2, 1);

 substring
-----------
 62
```

L'esempio seguente restituisce il nome `Ana` che appare dopo l'ultimo spazio nella stringa di input `Silva, Ana`.

```
select reverse(substring(reverse('Silva, Ana'), 1, position(' ' IN reverse('Silva,
 Ana'))))

 reverse
-----------
 Ana
```

Funzione TEXTLEN

Sinonimo della funzione LEN.

Per informazioni, consultare Funzione LEN.

Funzione TRANSLATE

Per una data espressione, sostituisce tutte le occorrenze di caratteri specificati con sostituti specificati. I caratteri esistenti sono mappati per i caratteri sostitutivi in base alla loro posizione negli argomenti characters_to_replace e characters_to_substitute. Se vengono specificati più caratteri nell'argomento characters_to_replace rispetto all'argomento characters_to_substitute, i caratteri extra dall'argomento characters_to_replace vengono omessi nel valore di restituzione.

TRANSLATE è simile a Funzione REPLACE e a Funzione REGEXP_REPLACE, ad eccezione del fatto che REPLACE sostituisce un'intera stringa con un'altra stringa e REGEXP_REPLACE consente

di cercare una stringa per un modello di espressione regolare, mentre TRANSLATE realizza più sostituzioni a carattere singolo.

Se qualsiasi argomento è null, il risultato è NULL.

Sintassi

```
TRANSLATE( expression, characters_to_replace, characters_to_substitute )
```

Argomenti

espressione

L'espressione da tradurre.

characters_to_replace

Una stringa contenente i caratteri da sostituire.

characters_to_substitute

Una stringa contenente i caratteri da sostituire.

Tipo restituito

VARCHAR

Esempi

Per sostituire vari caratteri in una stringa, utilizza l'esempio seguente.

```
SELECT TRANSLATE('mint tea', 'inea', 'osin');

+-----------+
| translate |
+-----------+
| most tin  |
+-----------+
```

Negli esempi seguenti vengono utilizzati i dati della tabella USERS database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per sostituire il simbolo at (@) con un punto per tutti i valori in una colonna, utilizza l'esempio seguente.

```
SELECT email, TRANSLATE(email, '@', '.') as obfuscated_email
FROM users LIMIT 10;

+--------------------------------------+--------------------------------------+
|                email                 |           obfuscated_email           |
+--------------------------------------+--------------------------------------+
| Cum@accumsan.com                     | Cum.accumsan.com                     |
| lorem.ipsum@Vestibulumante.com       | lorem.ipsum.Vestibulumante.com       |
| non.justo.Proin@ametconsectetuer.edu | non.justo.Proin.ametconsectetuer.edu |
| non.ante.bibendum@porttitortellus.org | non.ante.bibendum.porttitortellus.org |
| eros@blanditatnisi.org               | eros.blanditatnisi.org               |
| augue@Donec.ca                       | augue.Donec.ca                       |
| cursus@pedeacurna.edu                | cursus.pedeacurna.edu                |
| at@Duis.com                          | at.Duis.com                          |
| quam@facilisisvitaeorci.ca           | quam.facilisisvitaeorci.ca           |
| mi.lorem@nunc.edu                    | mi.lorem.nunc.edu                    |
+--------------------------------------+--------------------------------------+
```

Per sostituire gli spazi con caratteri di sottolineatura ed eliminare i punti per tutti i valori in una colonna, utilizza l'esempio seguente.

```
SELECT city, TRANSLATE(city, ' .', '_')
FROM users
WHERE city LIKE 'Sain%' OR city LIKE 'St%'
GROUP BY city
ORDER BY city;

+---------------+---------------+
|      city     |   translate   |
+---------------+---------------+
| Saint Albans  | Saint_Albans  |
| Saint Cloud   | Saint_Cloud   |
| Saint Joseph  | Saint_Joseph  |
| Saint Louis   | Saint_Louis   |
| Saint Paul    | Saint_Paul    |
| St. George    | St_George     |
| St. Marys     | St_Marys      |
| St. Petersburg | St_Petersburg |
| Stafford      | Stafford      |
| Stamford      | Stamford      |
```

```
| Stanton        | Stanton       |
| Starkville     | Starkville    |
| Statesboro     | Statesboro    |
| Staunton       | Staunton      |
| Steubenville   | Steubenville  |
| Stevens Point  | Stevens_Point |
| Stillwater     | Stillwater    |
| Stockton       | Stockton      |
| Sturgis        | Sturgis       |
+----------------+---------------+
```

Funzione TRIM

Taglia una stringa in base agli spazi vuoti o ai caratteri specificati.

Sintassi

```
TRIM( [ BOTH | LEADING | TRAILING ] [trim_chars FROM ] string )
```

Argomenti

BOTH | LEADING | TRAILING

(Facoltativo) Specifica da dove tagliare i caratteri. Utilizza BOTH per rimuovere i caratteri iniziali e finali, LEADING per rimuovere solo i caratteri iniziali e TRAILING per rimuovere solo i caratteri finali. Se questo parametro viene omesso, vengono tagliati sia i caratteri iniziali che quelli finali.

trim_chars

(Facoltativo) I caratteri da ridurre dalla stringa. Se questo parametro viene omesso, gli spazi vuoti vengono ridotti.

stringa

La stringa da ridurre.

Tipo restituito

La funzione TRIM restituisce una stringa `VARCHAR` o `CHAR`. Se utilizzi la funzione TRIM con un comando SQL, Amazon Redshift converte implicitamente i risultati in `VARCHAR`. Se si utilizza la funzione TRIM nell'elenco SELECT per una funzione SQL, Amazon Redshift non converte implicitamente i risultati e potrebbe essere necessario eseguire una conversione esplicita per

evitare un errore di mancata corrispondenza del tipo di dati. Consultare le funzioni Funzione CAST e Funzione CONVERT per informazioni sulle conversioni esplicite.

Esempi

Per tagliare gli spazi vuoti iniziali e finali dalla stringa dog , utilizza l'esempio seguente.

```
SELECT TRIM('    dog    ');

+-------+
| btrim |
+-------+
| dog   |
+-------+
```

Per tagliare gli spazi vuoti iniziali e finali dalla stringa dog , utilizza l'esempio seguente.

```
SELECT TRIM(BOTH FROM '    dog    ');

+-------+
| btrim |
+-------+
| dog   |
+-------+
```

Per rimuovere le virgolette doppie iniziali dalla stringa "dog", utilizza l'esempio seguente.

```
SELECT TRIM(LEADING '"' FROM'"dog"');

+-------+
| ltrim |
+-------+
| dog"  |
+-------+
```

Per rimuovere le virgolette doppie finali dalla stringa "dog", utilizza l'esempio seguente.

```
SELECT TRIM(TRAILING '"' FROM'"dog"');

+-------+
| rtrim |
```

```
+-------+
| "dog  |
+-------+
```

TRIM rimuove tutti i caratteri in trim_chars se questi si trovano all'inizio o alla fine della stringa. L'esempio seguente ritaglia i caratteri "C", "D" e "G" quando si trovano all'inizio o alla fine di VENUENAME che è una colonna VARCHAR. Per ulteriori informazioni, consulta Tabella VENUE.

```
SELECT venueid, venuename, TRIM('CDG' FROM venuename)
FROM venue
WHERE venuename LIKE '%Park'
ORDER BY 2
LIMIT 7;

+---------+----------------------------+----------------------------+
| venueid |         venuename          |           btrim            |
+---------+----------------------------+----------------------------+
|     121 | AT&T Park                  | AT&T Park                  |
|     109 | Citizens Bank Park         | itizens Bank Park          |
|     102 | Comerica Park              | omerica Park               |
|       9 | Dick's Sporting Goods Park | ick's Sporting Goods Park  |
|      97 | Fenway Park                | Fenway Park                |
|     112 | Great American Ball Park   | reat American Ball Park    |
|     114 | Miller Park                | Miller Park                |
+---------+----------------------------+----------------------------+
```

Funzione UPPER

Converte una stringa in maiuscolo. UPPER supporta caratteri multibyte UTF-8, fino a un massimo di quattro byte per carattere.

Sintassi

```
UPPER(string)
```

Argomenti

stringa

Il parametro di input è una stringa VARCHAR o qualsiasi altro tipo di dati, ad esempio CHAR, che è possibile convertire implicitamente in VARCHAR.

Tipo restituito

La funzione UPPER restituisce una stringa di caratteri che appartiene allo stesso tipo di dati della stringa di input. Ad esempio, se l'input è una stringa VARCHAR, la funzione restituirà una stringa VARCHAR.

Esempi

Nell'esempio seguente vengono utilizzati i dati della tabella CATEGORY database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

Per convertire il campo CATNAME in lettere maiuscole, utilizza l'esempio seguente.

```
SELECT catname, UPPER(catname)
FROM category
ORDER BY 1,2;

+-----------+-----------+
|  catname  |   upper   |
+-----------+-----------+
| Classical | CLASSICAL |
| Jazz      | JAZZ      |
| MLB       | MLB       |
| MLS       | MLS       |
| Musicals  | MUSICALS  |
| NBA       | NBA       |
| NFL       | NFL       |
| NHL       | NHL       |
| Opera     | OPERA     |
| Plays     | PLAYS     |
| Pop       | POP       |
+-----------+-----------+
```

Funzioni di informazioni sul tipo SUPER

Di seguito, è riportata una descrizione per le funzioni di informazioni sul tipo per SQL supportate da Amazon Redshift per ricavare le informazioni dinamiche dagli input del tipo di dati SUPER.

Argomenti

- Funzione DECIMAL_PRECISION
- Funzione DECIMAL_SCALE

- Funzione IS_ARRAY
- Funzione IS_BIGINT
- Funzione IS_BOOLEAN
- Funzione IS_CHAR
- Funzione IS_DECIMAL
- Funzione IS_FLOAT
- Funzione IS_INTEGER
- Funzione IS_OBJECT
- Funzione IS_SCALAR
- Funzione IS_SMALLINT
- Funzione IS_VARCHAR
- Funzione JSON_SIZE
- Funzione JSON_TYPEOF
- SIZE

Funzione DECIMAL_PRECISION

Controlla la precisione del numero totale massimo di cifre decimali da memorizzare. Questo numero include le cifre sia a sinistra che a destra della virgola decimale. L'intervallo di precisione è compreso tra 1 e 38, con un valore di default di 38.

Sintassi

```
DECIMAL_PRECISION(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

`INTEGER`

Esempi

Per applicare la funzione DECIMAL_PRECISION alla tabella t, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES (3.14159);

SELECT DECIMAL_PRECISION(s) FROM t;

+-------------------+
| decimal_precision |
+-------------------+
|                 6 |
+-------------------+
```

Funzione DECIMAL_SCALE

Controlla il numero di cifre decimali da memorizzare a destra del punto decimale. L'intervallo di precisione è compreso tra 0 e il punto di precisione, con un valore di default di 0.

Sintassi

```
DECIMAL_SCALE(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

INTEGER

Esempi

Per applicare la funzione DECIMAL_SCALE alla tabella t, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);
```

```
INSERT INTO t VALUES (3.14159);

SELECT DECIMAL_SCALE(s) FROM t;

+---------------+
| decimal_scale |
+---------------+
|             5 |
+---------------+
```

Funzione IS_ARRAY

Controlla se una variabile è un array. La funzione restituisce `true` se la variabile è un array. La funzione include anche array vuoti. In caso contrario, la funzione restituisce `false` per tutti gli altri valori, incluso null.

Sintassi

```
IS_ARRAY(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se `[1,2]` è un array utilizzando la funzione IS_ARRAY, utilizza l'esempio seguente.

```
SELECT IS_ARRAY(JSON_PARSE('[1,2]'));

+----------+
| is_array |
+----------+
| true     |
```

```
+----------+
```

Funzione IS_BIGINT

Controlla se un valore è di tipo `BIGINT`. La funzione IS_BIGINT restituisce `true` per i numeri di precisione 0 nell'intervallo a 64 bit. In caso contrario, la funzione restituisce `false` per tutti gli altri valori, inclusi i valori null e a virgola mobile.

La funzione IS_BIGINT è un superset di IS_INTEGER.

Sintassi

```
IS_BIGINT(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna `SUPER`.

Tipo restituito

`BOOLEAN`

Esempi

Per verificare se `5` è un valore `BIGINT` utilizzando la funzione IS_BIGINT, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES (5);

SELECT s, IS_BIGINT(s) FROM t;

+---+-----------+
| s | is_bigint |
+---+-----------+
| 5 | true      |
+---+-----------+
```

Funzione IS_BOOLEAN

Controlla se un valore è di tipo BOOLEAN. La funzione IS_BOOLEAN restituisce `true` per i valori booleani JSON costanti. La funzione restituisce `false` per qualsiasi altro valore, incluso null.

Sintassi

```
IS_BOOLEAN(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se TRUE è un valore BOOLEAN utilizzando la funzione IS_BOOLEAN, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES (TRUE);

SELECT s, IS_BOOLEAN(s) FROM t;

+------+------------+
|  s   | is_boolean |
+------+------------+
| true | true       |
+------+------------+
```

Funzione IS_CHAR

Controlla se un valore è di tipo CHAR. La funzione IS_CHAR restituisce `true` per le stringhe che contengono solo caratteri ASCII, poiché il tipo CHAR può archiviare solo caratteri in formato ASCII. La funzione restituisce `false` per qualsiasi altro valore.

Sintassi

```
IS_CHAR(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se `t` è un valore CHAR utilizzando la funzione IS_CHAR, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES ('t');

SELECT s, IS_CHAR(s) FROM t;

+-----+---------+
|  s  | is_char |
+-----+---------+
| "t" | true    |
+-----+---------+
```

Funzione IS_DECIMAL

Controlla se un valore è di tipo `DECIMAL`. La funzione IS_DECIMAL restituisce `true` per i numeri che non sono a virgola mobile. La funzione restituisce `false` per qualsiasi altro valore, incluso null.

La funzione IS_DECIMAL è un superset di IS_BIGINT.

Sintassi

```
IS_DECIMAL(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se 1.22 è un valore DECIMAL utilizzando la funzione IS_DECIMAL, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES (1.22);

SELECT s, IS_DECIMAL(s) FROM t;

+------+------------+
|  s   | is_decimal |
+------+------------+
| 1.22 | true       |
+------+------------+
```

Funzione IS_FLOAT

Controlla se un valore è un numero a virgola mobile. La funzione IS_FLOAT restituisce `true` per i numeri a virgola mobile (FLOAT4 e FLOAT8). La funzione restituisce `false` per qualsiasi altro valore.

Il set IS_DECIMAL e il set IS_FLOAT sono separati.

Sintassi

```
IS_FLOAT(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se 2.22::FLOAT è un valore FLOAT utilizzando la funzione IS_FLOAT, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES(2.22::FLOAT);

SELECT s, IS_FLOAT(s) FROM t;

+---------+----------+
|    s    | is_float |
+---------+----------+
| 2.22e+0 | true     |
+---------+----------+
```

Funzione IS_INTEGER

Restituisce `true` per i numeri di precisione 0 nell'intervallo a 32 bit e `false` per qualsiasi altro valore (inclusi i valori null e a virgola mobile).

La funzione IS_INTEGER è un superset della funzione IS_SMALLINT.

Sintassi

```
IS_INTEGER(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se 5 è un valore INTEGER utilizzando la funzione IS_INTEGER, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES (5);

SELECT s, IS_INTEGER(s) FROM t;

+---+------------+
| s | is_integer |
+---+------------+
| 5 | true       |
+---+------------+
```

Funzione IS_OBJECT

Controlla se una variabile è un oggetto. La funzione IS_OBJECT restituisce `true` per gli oggetti, inclusi gli oggetti vuoti. La funzione restituisce `false` per qualsiasi altro valore, incluso null.

Sintassi

```
IS_OBJECT(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se `{"name": "Joe"}` è un oggetto utilizzando la funzione IS_OBJECT, utilizza l'esempio seguente.

```
CREATE TABLE t(s super);
```

```
INSERT INTO t VALUES (JSON_PARSE('{"name": "Joe"}'));

SELECT s, IS_OBJECT(s) FROM t;

+----------------+-----------+
|       s        | is_object |
+----------------+-----------+
| {"name":"Joe"} | true      |
+----------------+-----------+
```

Funzione IS_SCALAR

Controlla se una variabile è un oggetto scalare. La funzione IS_SCALAR restituisce `true` per qualsiasi valore che non è un array o un oggetto. La funzione restituisce `false` per qualsiasi altro valore, incluso null.

Il set IS_ARRAY, IS_OBJECT e IS_SCALAR copre tutti i valori tranne i valori di tipo null.

Sintassi

```
IS_SCALAR(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se `{"name": "Joe"}` è un valore scalare utilizzando la funzione IS_SCALAR, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES (JSON_PARSE('{"name": "Joe"}'));
```

```
SELECT s, IS_SCALAR(s.name) FROM t;

+----------------+-----------+
|       s        | is_scalar |
+----------------+-----------+
| {"name":"Joe"} | true      |
+----------------+-----------+
```

Funzione IS_SMALLINT

Controlla se una variabile è SMALLINT. La funzione IS_SMALLINT restituisce `true` per i numeri di precisione 0 nell'intervallo a 16 bit. La funzione restituisce `false` per tutti gli altri valori, inclusi i valori null e a virgola mobile.

Sintassi

```
IS_SMALLINT(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Return

BOOLEAN

Esempi

Per verificare se 5 è un valore SMALLINT utilizzando la funzione IS_SMALLINT, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES (5);

SELECT s, IS_SMALLINT(s) FROM t;

+---+-------------+
| s | is_smallint |
```

```
+---+-------------+
| 5 | true        |
+---+-------------+
```

Funzione IS_VARCHAR

Controlla se una variabile è VARCHAR. La funzione IS_VARCHAR restituisce `true` per tutte le stringhe. La funzione restituisce `false` per qualsiasi altro valore.

La funzione IS_VARCHAR è un superset della funzione IS_CHAR.

Sintassi

```
IS_VARCHAR(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

BOOLEAN

Esempi

Per verificare se `abc` è un valore VARCHAR utilizzando la funzione IS_VARCHAR, utilizza l'esempio seguente.

```
CREATE TABLE t(s SUPER);

INSERT INTO t VALUES ('abc');

SELECT s, IS_VARCHAR(s) FROM t;

+-------+------------+
|   s   | is_varchar |
+-------+------------+
| "abc" | true       |
+-------+------------+
```

Funzione JSON_SIZE

La funzione JSON_SIZE restituisce il numero di byte nell'espressione SUPER specificata quando viene serializzata in una stringa.

Sintassi

```
JSON_SIZE(super_expression)
```

Argomenti

super_expression

Una costante o un'espressione SUPER.

Tipo restituito

INTEGER

La funzione JSON_SIZE restituisce un valore INTEGER che indica il numero di byte nella stringa di input. Questo valore è diverso dal numero di caratteri. Ad esempio, il carattere UTF-8 # (punto nero) ha una dimensione di 3 byte anche se è un solo carattere.

Note per l'utilizzo

JSON_SIZE(x) è funzionalmente identica a OCTET_LENGTH(JSON_SERIALIZE). Tuttavia, nota che JSON_SERIALIZE restituisce un errore quando l'espressione SUPER fornita supera il limite VARCHAR del sistema quando serializzata. JSON_SIZE non ha questa limitazione.

Esempi

Per restituire la lunghezza di un valore SUPER serializzato su una stringa, utilizza l'esempio seguente.

```
SELECT JSON_SIZE(JSON_PARSE('[10001,10002,"#"]'));

+-----------+
| json_size |
+-----------+
|        19 |
+-----------+
```

Tieni presente che l'espressione SUPER fornita è lunga 17 caratteri, ma ¨# è un carattere a 3 byte, quindi JSON_SIZE restituisce 19.

Funzione JSON_TYPEOF

La funzione scalare JSON_TYPEOF restituisce un VARCHAR con valori boolean, number, string, object, array o null, a seconda del tipo dinamico del valore SUPER.

Sintassi

```
JSON_TYPEOF(super_expression)
```

Argomenti

super_expression

Un'espressione o una colonna SUPER.

Tipo restituito

VARCHAR

Esempi

Per verificare il tipo di JSON per l'array `[1,2]` utilizzando la funzione JSON_TYPEOF, utilizza l'esempio seguente.

```
SELECT JSON_TYPEOF(ARRAY(1,2));

+-------------+
| json_typeof |
+-------------+
| array       |
+-------------+
```

Per verificare il tipo di JSON per l'oggetto `{"name":"Joe"}` utilizzando la funzione JSON_TYPEOF, utilizza l'esempio seguente.

```
SELECT JSON_TYPEOF(JSON_PARSE('{"name":"Joe"}'));

+-------------+
```

```
| json_typeof |
+-------------+
| object      |
+-------------+
```

SIZE

Restituisce la dimensione binaria in memoria di una costante o espressione di tipo SUPER come INTEGER.

Sintassi

```
SIZE(super_expression)
```

Argomenti

super_expression

Una costante o un'espressione di tipo SUPER.

Tipo restituito

INTEGER

Esempi

Per utilizzare SIZE per ottenere la dimensione in memoria di diverse espressioni di tipo SUPER, utilizza l'esempio seguente.

```
CREATE TABLE test_super_size(a SUPER);

INSERT INTO test_super_size
VALUES
  (null),
  (TRUE),
  (JSON_PARSE('[0,1,2,3]')),
  (JSON_PARSE('{"a":0,"b":1,"c":2,"d":3}'))
;

SELECT a, SIZE(a)
FROM test_super_size
ORDER BY 2, 1;
```

```
+---------------------------+------+
|             a             | size |
+---------------------------+------+
| true                      |    4 |
| NULL                      |    4 |
| [0,1,2,3]                 |   23 |
| {"a":0,"b":1,"c":2,"d":3} |   52 |
+---------------------------+------+
```

Operatori e funzioni VARBYTE

Le funzioni e gli operatori di Amazon Redshift che supportano il tipo di dati VARBYTE includono:

- Operatori VARBYTE
- FROM_HEX
- FROM_VARBYTE
- GETBIT
- TO_HEX
- TO_VARBYTE
- CONCAT
- LEN
- Funzione LENGTH
- OCTET_LENGTH
- Funzione SUBSTRING

Operatori VARBYTE

Nella tabella seguente sono elencati gli operatori VARBYTE. L'operatore utilizza valore binario del tipo di dati VARBYTE. Se uno o entrambi gli input sono nulli, il risultato è nullo.

Operatori supportati

Operatore	Descrizione	Tipo restituito
<	Minore di	BOOLEAN

Operatore	Descrizione	Tipo restituito
<=	Minore di o uguale a	BOOLEAN
=	Uguale	BOOLEAN
>	Maggiore di	BOOLEAN
>=	Maggiore di o uguale a	BOOLEAN
!= or <>	Non uguale	BOOLEAN
||	Concatenazione	VARBYTE
+	Concatenazione	VARBYTE
~	Non bitwise	VARBYTE
&	Bitwise e	VARBYTE
|	Bitwise o	VARBYTE
#	Bitwise xor	VARBYTE

Esempi

Negli esempi seguenti, il valore di `'a'::VARBYTE` è 61 e il valore di `'b'::VARBYTE` è 62. `::` converte le stringhe nel tipo di dati VARBYTE. Per ulteriori informazioni sulla conversione dei tipi di dati, consulta CAST.

Per confrontare se `'a'` è inferiore a `'b'` utilizzando l'operatore <, utilizza l'esempio seguente.

```
SELECT 'a'::VARBYTE < 'b'::VARBYTE AS less_than;

+-----------+
```

```
| less_than |
+-----------+
| true      |
+-----------+
```

Per confrontare se `'a'` equivale a `'b'` utilizzando l'operatore =, utilizza l'esempio seguente..

```
SELECT 'a'::VARBYTE = 'b'::VARBYTE AS equal;

+-------+
| equal |
+-------+
| false |
+-------+
```

Per concatenare due valori binari utilizzando l'operatore `||`, utilizza l'esempio seguente.

```
SELECT 'a'::VARBYTE || 'b'::VARBYTE AS concat;

+--------+
| concat |
+--------+
|   6162 |
+--------+
```

Per concatenare due valori binari utilizzando l'operatore +, utilizza l'esempio seguente.

```
SELECT 'a'::VARBYTE + 'b'::VARBYTE AS concat;

+--------+
| concat |
+--------+
|   6162 |
+--------+
```

Per negare ogni bit del valore binario di input utilizzando la funzione FROM_VARBYTE, utilizza l'esempio seguente. La stringa `'a'` restituisce `01100001`. Per ulteriori informazioni, consulta FROM_VARBYTE.

```
SELECT FROM_VARBYTE(~'a'::VARBYTE, 'binary');
```

```
+--------------+
| from_varbyte |
+--------------+
|     10011110 |
+--------------+
```

Per applicare l'operatore & sui due valori binari di input, utilizza l'esempio seguente. La stringa `'a'` restituisce `01100001` e `'b'` restituisce `01100010`.

```
SELECT FROM_VARBYTE('a'::VARBYTE & 'b'::VARBYTE, 'binary');

+--------------+
| from_varbyte |
+--------------+
|     01100000 |
+--------------+
```

funzione FROM_HEX

FROM_HEX converte un esadecimale in un valore binario.

Sintassi

```
FROM_HEX(hex_string)
```

Argomenti

hex_string

Stringa esadecimale di tipo di dati `VARCHAR` o `TEXT` da convertire. Il formato deve essere un valore letterale.

Tipo restituito

`VARBYTE`

Esempi

Per convertire la rappresentazione esadecimale di `'6162'` in un valore binario, utilizza l'esempio seguente. Il risultato viene visualizzato automaticamente come rappresentazione esadecimale del valore binario.

```
SELECT FROM_HEX('6162');

+----------+
| from_hex |
+----------+
|     6162 |
+----------+
```

funzione FROM_VARBYTE

FROM_VARBYTE converte un valore binario in una stringa di caratteri nel formato specificato.

Sintassi

```
FROM_VARBYTE(binary_value, format)
```

Argomenti

binary_value

Un valore binario di tipo di dati VARBYTE.

format

Il formato della stringa di caratteri restituita. I valori validi senza distinzione tra maiuscole e minuscole sono `hex`, `binary`, `utf8` (anche `utf-8` e `utf_8`) e `base64`.

Tipo restituito

VARCHAR

Esempi

Per convertire il valore binario `'ab'` in esadecimale, utilizza l'esempio seguente.

```
SELECT FROM_VARBYTE('ab', 'hex');

+--------------+
| from_varbyte |
+--------------+
|         6162 |
+--------------+
```

Per restituire la rappresentazione binaria di `'4d'`, utilizza l'esempio seguente. La rappresentazione binaria di `'4d'` è la stringa di caratteri `01001101`.

```
SELECT FROM_VARBYTE(FROM_HEX('4d'), 'binary');

+--------------+
| from_varbyte |
+--------------+
|     01001101 |
+--------------+
```

Funzione GETBIT

GETBIT restituisce il valore in bit di un valore binario all'indice specificato.

Sintassi

```
GETBIT(binary_value, index)
```

Argomenti

binary_value

Un valore binario di tipo di dati VARBYTE.

indice

Un numero di indice del bit nel valore binario restituito. Il valore binario è un array di bit basato su 0 che viene indicizzato dal bit più a destra (bit meno significativo) al bit più a sinistra (bit più significativo).

Tipo restituito

`INTEGER`

Esempi

Per restituire il bit all'indice 2 del valore binario `from_hex('4d')`, utilizza l'esempio seguente. La rappresentazione binaria di `'4d'` è `01001101`.

```
SELECT GETBIT(FROM_HEX('4d'), 2);
```

```
+--------+
| getbit |
+--------+
|      1 |
+--------+
```

Per restituire il bit in otto posizioni dell'indice del valore binario restituito da `from_hex('4d')`, utilizza l'esempio seguente. La rappresentazione binaria di `'4d'` è `01001101`.

```
SELECT GETBIT(FROM_HEX('4d'), 7), GETBIT(FROM_HEX('4d'), 6),
  GETBIT(FROM_HEX('4d'), 5), GETBIT(FROM_HEX('4d'), 4),
  GETBIT(FROM_HEX('4d'), 3), GETBIT(FROM_HEX('4d'), 2),
  GETBIT(FROM_HEX('4d'), 1), GETBIT(FROM_HEX('4d'), 0);

+--------+--------+--------+--------+--------+--------+--------+--------+
| getbit | getbit | getbit | getbit | getbit | getbit | getbit | getbit |
+--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |      1 |      0 |      0 |      1 |      1 |      0 |      1 |
+--------+--------+--------+--------+--------+--------+--------+--------+
```

Funzione TO_HEX

TO_HEX converte un valore numerico o binario in una rappresentazione esadecimale.

Sintassi

```
TO_HEX(value)
```

Argomenti

value

Un valore numerico o binario (VARBYTE) da convertire.

Tipo restituito

VARCHAR

Esempi

Per convertire un numero nella sua rappresentazione esadecimale, utilizza l'esempio seguente.

```
SELECT TO_HEX(2147676847);

+----------+
|  to_hex  |
+----------+
| 8002f2af |
+----------+
```

Per convertire la rappresentazione VARBYTE di `'abc'` in un numero esadecimale, utilizza l'esempio seguente.

```
SELECT TO_HEX('abc'::VARBYTE);

+--------+
| to_hex |
+--------+
| 616263 |
+--------+
```

Per creare una tabella, inserire la rappresentazione VARBYTE di `'abc'` in un numero esadecimale e quindi selezionare la colonna con il valore, utilizza l'esempio seguente.

```
CREATE TABLE t (vc VARCHAR);
INSERT INTO t SELECT TO_HEX('abc'::VARBYTE);
SELECT vc FROM t;

+--------+
|   vc   |
+--------+
| 616263 |
+--------+
```

Per mostrare che quando si converte un valore VARBYTE in VARCHAR il formato è UTF-8, utilizza l'esempio seguente.

```
CREATE TABLE t (vc VARCHAR);
INSERT INTO t SELECT 'abc'::VARBYTE::VARCHAR;

SELECT vc FROM t;

+-----+
```

```
| vc  |
+-----+
| abc |
+-----+
```

Funzione TO_VARBYTE

TO_VARBYTE converte una stringa in un formato specificato in un valore binario.

Sintassi

```
TO_VARBYTE(string, format)
```

Argomenti

stringa

Una stringa CHAR o VARCHAR.

format

Il formato della stringa di input. I valori validi senza distinzione tra maiuscole e minuscole sono `hex`, `binary`, `utf8` (anche `utf-8` e `utf_8`) e `base64`.

Tipo restituito

VARBYTE

Esempi

Per convertire l'esadecimale 6162 in un valore binario, utilizza l'esempio seguente. Il risultato viene visualizzato automaticamente come rappresentazione esadecimale del valore binario.

```
SELECT TO_VARBYTE('6162', 'hex');

+------------+
| to_varbyte |
+------------+
|       6162 |
+------------+
```

Per restituire la rappresentazione binaria di 4d, utilizza l'esempio seguente. La rappresentazione binaria di '4d' è 01001101.

```
SELECT TO_VARBYTE('01001101', 'binary');

+------------+
| to_varbyte |
+------------+
|         4d |
+------------+
```

Per convertire la stringa 'a' in UTF-8 in un valore binario, utilizza l'esempio seguente. Il risultato viene visualizzato automaticamente come rappresentazione esadecimale del valore binario.

```
SELECT TO_VARBYTE('a', 'utf8');

+------------+
| to_varbyte |
+------------+
|         61 |
+------------+
```

Per convertire la stringa '4' in esadecimale in un valore binario, utilizza l'esempio seguente. Se la lunghezza della stringa esadecimale è un numero dispari, allora un 0 è presupposto per formare un numero esadecimale valido.

```
SELECT TO_VARBYTE('4', 'hex');

+------------+
| to_varbyte |
+------------+
|         04 |
+------------+
```

Funzioni finestra

Le funzioni finestra ti consentono di creare query aziendali analitiche in modo più efficiente. Le funzioni finestra operano su una partizione o "finestra" di un insieme di risultati e restituiscono un valore per ogni riga in quella finestra. Tuttavia, le funzioni non finestra eseguono i calcoli in relazione

a ogni riga del set di risultati. A differenza delle funzioni di gruppo che aggregano le righe dei risultati, le funzioni finestra mantengono tutte le righe nell'espressione della tabella.

I valori restituiti sono calcolati utilizzando i valori dai set di righe in quella finestra. Per ogni riga nella tabella, la finestra definisce un set di righe che viene utilizzato per calcolare gli attributi aggiuntivi. Una finestra viene definita utilizzando una specifica della finestra (la clausola OVER) e si basa su tre concetti principali:

- Partizionamento della finestra, che forma gruppi di righe (clausola PARTITION)
- Ordinamento della finestra, che definisce un ordine o una sequenza di righe all'interno di ciascuna partizione (clausola ORDER BY)
- Frame della finestra, che sono definiti in relazione a ciascuna riga per restringere ulteriormente l'insieme di righe (specifica ROWS)

Le funzioni finestra sono l'ultimo insieme di operazioni eseguite in una query ad eccezione della clausola ORDER BY finale. Tutti i join e tutte le clausole WHERE, GROUP BY e HAVING vengono completati prima che le funzioni finestra vengano elaborate. Pertanto, le funzioni finestra possono essere visualizzate solo nell'elenco di selezione o nella clausola ORDER BY. È possibile utilizzare più funzioni finestra all'interno di una singola query con diverse clausole del frame. È inoltre possibile utilizzare le funzioni finestra in altre espressioni scalari, come ad esempio CASE.

Riepilogo della sintassi della funzione finestra

Le funzioni della finestra seguono una sintassi standard, che è la seguente.

```
function (expression) OVER (
[ PARTITION BY expr_list ]
[ ORDER BY order_list [ frame_clause ] ] )
```

Qui, function è una delle funzioni descritte in questa sezione.

L'expr_list è il seguente.

```
expression | column_name [, expr_list ]
```

L'order_list è il seguente.

```
expression | column_name [ ASC | DESC ]
```

```
[ NULLS FIRST | NULLS LAST ]
[, order_list ]
```

La frame_clause è la seguente.

```
ROWS
{ UNBOUNDED PRECEDING | unsigned_value PRECEDING | CURRENT ROW } |

{ BETWEEN
{ UNBOUNDED PRECEDING | unsigned_value { PRECEDING | FOLLOWING } | CURRENT ROW}
AND
{ UNBOUNDED FOLLOWING | unsigned_value { PRECEDING | FOLLOWING } | CURRENT ROW }}
```

Argomenti

funzione

La funzione finestra. Per informazioni dettagliate, vedere le descrizioni della singola funzione.

OVER

La clausola che definisce la specifica della finestra. La clausola OVER è obbligatoria per le funzioni finestra e le contraddistingue da altre funzioni SQL.

PARTITION BY expr_list

(Facoltativo) La clausola PARTITION BY suddivide il set di risultati in partizioni in modo molto simile alla clausola GROUP BY. Se è presente una clausola di partizione, la funzione viene calcolata per le righe in ogni partizione. Se non viene specificata alcuna clausola di partizione, una singola partizione contiene l'intera tabella e la funzione viene calcolata per quella tabella completa.

Le funzioni di classificazione DENSE_RANK, NTILE, RANK e ROW_NUMBER richiedono un confronto globale di tutte le righe nell'insieme di risultati. Quando viene utilizzata una clausola PARTITION BY, l'ottimizzatore della query può eseguire ciascuna aggregazione in parallelo distribuendo il carico di lavoro su più sezioni in base alle partizioni. Se la clausola PARTITION BY non è presente, la fase di aggregazione deve essere eseguita in serie su una singola sezione, che può avere un impatto negativo significativo sulle prestazioni, in particolare per i cluster di grandi dimensioni.

Amazon Redshift non supporta letterali stringa nelle clausole PARTITION BY.

ORDER BY order_list

(Facoltativo) La funzione finestra viene applicata alle righe all'interno di ciascuna partizione ordinata in base alla specifica dell'ordine in ORDER BY. Questa clausola ORDER BY è distinta e completamente non correlata a una clausola ORDER BY in una frame_clause. La clausola ORDER BY può essere utilizzata senza la clausola PARTITION BY.

Per le funzioni di classificazione, la clausola ORDER BY identifica le misure per i valori di classificazione. Per le funzioni di aggregazione, le righe partizionate devono essere ordinate prima che la funzione di aggregazione sia calcolata per ciascun frame. Per ulteriori informazioni sui tipi di funzione finestra, consultare Funzioni finestra.

Gli identificatori o le espressioni di colonna che valutano gli identificatori di colonna sono obbligatori nell'elenco degli ordini. Né le costanti né le espressioni costanti possono essere utilizzate come sostituti dei nomi delle colonne.

I valori NULL vengono trattati come il proprio gruppo, ordinati e classificati in base all'opzione NULLS FIRST o NULLS LAST. Per impostazione predefinita, i valori NULL vengono ordinati e classificati per ultimi in ordine ASC e ordinati e classificati per primi in ordine DESC.

Amazon Redshift non supporta letterali stringa nelle clausole ORDER BY.

Se viene omessa la clausola ORDER BY, l'ordine delle righe non è deterministico.

> **Note**
>
> In qualsiasi sistema parallelo come ad esempio Amazon Redshift, quando una clausola ORDER BY non produce un ordinamento univoco e totale dei dati, l'ordine delle righe è non deterministico. Ovvero, se l'espressione ORDER BY produce valori duplicati (un ordinamento parziale), l'ordine di restituzione di tali righe potrebbe variare da una sola esecuzione di Amazon Redshift a quella successiva. A loro volta, le funzioni finestra potrebbero restituire risultati inattesi o incoerenti. Per ulteriori informazioni, consultare Ordinamento univoco dei dati per le funzioni finestra.

column_name

Nome di una colonna da partizionare o da ordinare.

ASC | DESC

Opzione che definisce l'ordinamento per l'espressione, come segue:

- ASC: crescente (ad esempio, dal più piccolo al più grande per i valori numerici e da 'A' a 'Z' per le stringhe di caratteri). Se non viene specificata alcuna opzione, i dati vengono ordinati in ordine crescente per impostazione predefinita.
- DESC: decrescente (ad esempio, dal più grande al più piccolo per i valori numerici e da 'Z' ad 'A' per le stringhe).

NULLS FIRST | NULLS LAST

Opzione che specifica se NULLS deve essere ordinato per primo, prima di valori non null, o per ultimo, dopo valori non null. Per impostazione predefinita, i NULLS vengono ordinati e classificati per ultimi in ordine ASC e ordinati e classificati per primi in ordine DESC.

frame_clause

Per le funzioni di aggregazione, la clausola frame perfeziona ulteriormente l'insieme di righe in una finestra della funzione quando si utilizza ORDER BY. Fornisce la capacità di includere o escludere set di righe all'interno del risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati.

La clausola frame non si applica alle funzioni di classificazione. Inoltre, la clausola frame non è richiesta quando non viene utilizzata alcuna clausola ORDER BY nella clausola OVER per una funzione di aggregazione. Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita.

Quando non viene specificata alcuna clausola ORDER BY, il frame implicito è illimitato: equivalente a ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING.

ROWS

Questa clausola definisce il frame della finestra specificando una compensazione fisica dalla riga corrente.

Questa clausola specifica le righe nella finestra o partizione corrente con cui deve essere combinato il valore nella riga corrente. Usa argomenti che specificano la posizione della riga, che può essere prima o dopo la riga corrente. Il punto di riferimento per tutti i frame della finestra è la riga corrente. Ogni riga diventa a sua volta la riga corrente mentre il frame della finestra scorre in avanti nella partizione.

Il frame può essere un semplice insieme di righe fino a includere la riga corrente:

```
{UNBOUNDED PRECEDING | offset PRECEDING | CURRENT ROW}
```

Oppure può essere un insieme di righe tra due limiti.

```
BETWEEN
{ UNBOUNDED PRECEDING | offset { PRECEDING | FOLLOWING } | CURRENT ROW }
AND
{ UNBOUNDED FOLLOWING | offset { PRECEDING | FOLLOWING } | CURRENT ROW }
```

UNBOUNDED PRECEDING indica che la finestra inizia nella prima riga della partizione; offset PRECEDING indica che la finestra inizia un numero di righe equivalente al valore di compensazione prima della riga corrente. UNBOUNDED PRECEDING è il valore predefinito.

CURRENT ROW indica che la finestra inizia o termina nella riga corrente.

UNBOUNDED FOLLOWING indica che la finestra termina nell'ultima riga della partizione; offset FOLLOWING indica che la finestra termina un numero di righe equivalente al valore di compensazione dopo la riga corrente.

offset identifica un numero fisico di righe prima o dopo la riga corrente. In questo caso, offset deve essere una costante che valuta un valore numerico positivo. Ad esempio, 5 FOLLOWING terminerà il frame 5 righe dopo la riga corrente.

Laddove BETWEEN non viene specificato, il frame è implicitamente limitato dalla riga corrente. Ad esempio, `ROWS 5 PRECEDING` è uguale a `ROWS BETWEEN 5 PRECEDING AND CURRENT ROW`. Inoltre, `ROWS UNBOUNDED FOLLOWING` è uguale a `ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING`.

> **Note**
>
> Non è possibile specificare un frame in cui il limite iniziale è maggiore del limite finale. Ad esempio, non è possibile specificare nessuno di questi frame:
>
> ```
> between 5 following and 5 preceding
> between current row and 2 preceding
> between 3 following and current row
> ```

Ordinamento univoco dei dati per le funzioni finestra

Se una clausola ORDER BY per una funzione finestra non produce un ordinamento univoco e totale dei dati, l'ordine delle righe è non deterministico. Se l'espressione ORDER BY produce valori duplicati

(un ordinamento parziale), l'ordine di restituzione di tali righe può variare in più esecuzioni. In questo caso, le funzioni finestra possono restituire risultati inattesi o incoerenti.

Ad esempio, la seguente query restituisce risultati diversi su più esecuzioni. Si ottengono questi risultati diversi perché `order by dateid` non genera un ordinamento univoco dei dati per la funzione finestra SUM.

```
select dateid, pricepaid,
sum(pricepaid) over(order by dateid rows unbounded preceding) as sumpaid
from sales
group by dateid, pricepaid;

dateid | pricepaid |   sumpaid
--------+-----------+-------------
1827 |   1730.00 |     1730.00
1827 |    708.00 |     2438.00
1827 |    234.00 |     2672.00
...

select dateid, pricepaid,
sum(pricepaid) over(order by dateid rows unbounded preceding) as sumpaid
from sales
group by dateid, pricepaid;

dateid | pricepaid |   sumpaid
--------+-----------+-------------
1827 |    234.00 |      234.00
1827 |    472.00 |      706.00
1827 |    347.00 |     1053.00
...
```

In questo caso, l'aggiunta di una seconda colonna ORDER BY alla funzione finestra potrebbe risolvere il problema.

```
select dateid, pricepaid,
sum(pricepaid) over(order by dateid, pricepaid rows unbounded preceding) as sumpaid
from sales
group by dateid, pricepaid;

dateid | pricepaid | sumpaid
--------+-----------+---------
1827 |    234.00 |  234.00
```

```
1827 |    337.00 |  571.00
1827 |    347.00 |  918.00
...
```

Funzioni supportate

Amazon Redshift supporta due tipi di funzioni finestra: di aggregazione e di classificazione.

Queste sono le funzioni di aggregazione supportate:

- Funzione finestra AVG
- Funzione finestra COUNT
- Funzione finestra CUME_DIST
- Funzione finestra DENSE_RANK
- Funzione finestra FIRST_VALUE
- Funzione finestra LAG
- Funzione finestra LAST_VALUE
- Funzione finestra LEAD
- Funzione finestra LISTAGG
- Funzione finestra MAX
- Funzione finestra MEDIAN
- Funzione finestra MIN
- Funzione finestra NTH_VALUE
- Funzione finestra PERCENTILE_CONT
- Funzione finestra PERCENTILE_DISC
- Funzione finestra RATIO_TO_REPORT
- Funzioni finestra STDDEV_SAMP e STDDEV_POP (STDDEV_SAMP e STDDEV sono sinonimi)
- Funzione finestra SUM
- Funzioni finestra VAR_SAMP e VAR_POP (VAR_SAMP e VARIANCE sono sinonimi)

Queste sono le funzioni di classificazione supportate:

- Funzione finestra DENSE_RANK
- Funzione finestra NTILE

- Funzione finestra PERCENT_RANK
- Funzione finestra RANK
- Funzione finestra ROW_NUMBER

Tabella di esempio per gli esempi della funzione finestra

Sono presenti esempi di funzione finestra specifici con la descrizione di ogni funzione. Alcuni degli esempi utilizzano una tabella denominata WINSALES, che contiene 11 righe, come mostrato di seguito.

SALESID	DATEID	SELLERID	BUYERID	QTÀ	QTY_SHIPPED
30001	2/8/2003	3	B	10	10
10001	24/12/2003	1	C	10	10
10005	24/12/2003	1	A	30	
40001	9/1/2004	4	A	40	
10006	18/01/2004	1	C	10	
20001	12/2/2004	2	B	20	20
40005	12/2/2004	4	A	10	10
20002	16/2/2004	2	C	20	20
30003	18/4/2004	3	B	15	
30004	18/4/2004	3	B	20	
30007	7/9/2004	3	C	30	

Il seguente script crea e popola la tabella WINSALES di esempio.

```
CREATE TABLE winsales(
  salesid int,
```

```
  dateid date,
  sellerid int,
  buyerid char(10),
  qty int,
  qty_shipped int);

INSERT INTO winsales VALUES
  (30001, '8/2/2003', 3, 'b', 10, 10),
  (10001, '12/24/2003', 1, 'c', 10, 10),
  (10005, '12/24/2003', 1, 'a', 30, null),
  (40001, '1/9/2004', 4, 'a', 40, null),
  (10006, '1/18/2004', 1, 'c', 10, null),
  (20001, '2/12/2004', 2, 'b', 20, 20),
  (40005, '2/12/2004', 4, 'a', 10, 10),
  (20002, '2/16/2004', 2, 'c', 20, 20),
  (30003, '4/18/2004', 3, 'b', 15, null),
  (30004, '4/18/2004', 3, 'b', 20, null),
  (30007, '9/7/2004', 3, 'c', 30, null);
```

Funzione finestra AVG

La funzione finestra AVG restituisce la media (media aritmetica) dei valori di espressione di input. La funzione AVG funziona con i valori numerici e ignora i valori NULL.

Sintassi

```
AVG ( [ALL ] expression ) OVER
(
[ PARTITION BY expr_list ]
[ ORDER BY order_list
                        frame_clause ]
)
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

ALL

Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione per il conteggio. ALL è il valore predefinito. DISTINCT non è supportato.

OVER

Specifica le clausole finestra per le funzioni di aggregazione. La clausola OVER distingue le funzioni di aggregazione delle finestre dalle normali funzioni di aggregazione dell'insieme.

PARTITION BY expr_list

Definisce la finestra per la funzione AVG in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe all'interno del risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipi di dati

I tipi di argomenti supportati dalla funzione AVG sono SMALLINT, INTEGER, BIGINT, NUMERIC, DECIMAL, REAL e DOUBLE PRECISION.

I tipi di restituzione supportati dalla funzione AVG sono:

- BIGINT per gli argomenti SMALLINT oppure INTEGER
- NUMERIC per gli argomenti BIGINT
- DOUBLE PRECISION per argomenti del numero in virgola mobile

Esempi

Nel seguente esempio viene calcolata una media mobile delle quantità vendute per data; i risultati sono ordinati per ID data e ID vendite:

```
select salesid, dateid, sellerid, qty,
avg(qty) over
(order by dateid, salesid rows unbounded preceding) as avg
from winsales
```

```
order by 2,1;

salesid |   dateid   | sellerid | qty | avg
--------+------------+----------+-----+-----
30001 | 2003-08-02 |        3 |  10 |  10
10001 | 2003-12-24 |        1 |  10 |  10
10005 | 2003-12-24 |        1 |  30 |  16
40001 | 2004-01-09 |        4 |  40 |  22
10006 | 2004-01-18 |        1 |  10 |  20
20001 | 2004-02-12 |        2 |  20 |  20
40005 | 2004-02-12 |        4 |  10 |  18
20002 | 2004-02-16 |        2 |  20 |  18
30003 | 2004-04-18 |        3 |  15 |  18
30004 | 2004-04-18 |        3 |  20 |  18
30007 | 2004-09-07 |        3 |  30 |  19
(11 rows)
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Funzione finestra COUNT

La funzione finestra COUNT conta le righe definite dall'espressione.

La funzione COUNT ha due variazioni. COUNT(*) conta tutte le righe nella tabella di destinazione indipendentemente dal fatto che includano valori null o no. COUNT(espressione) calcola il numero di righe con valori non NULL in una colonna o espressione specifica.

Sintassi

```
COUNT ( * | [ ALL ] expression) OVER
(
[ PARTITION BY expr_list ]
[ ORDER BY order_list
                        frame_clause ]
)
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

ALL

Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione per il conteggio. ALL è il valore predefinito. DISTINCT non è supportato.

OVER

Specifica le clausole finestra per le funzioni di aggregazione. La clausola OVER distingue le funzioni di aggregazione delle finestre dalle normali funzioni di aggregazione dell'insieme.

PARTITION BY expr_list

Definisce la finestra per la funzione COUNT in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe all'interno del risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipi di dati

La funzione COUNT supporta tutti i tipi di dati degli argomenti.

Il tipo di restituzione supportato dalla funzione COUNT è BIGINT.

Esempi

Nel seguente esempio sono mostrati l'ID vendite, la quantità e il conteggio di tutte le righe dall'inizio della finestra dei dati:

```
select salesid, qty,
count(*) over (order by salesid rows unbounded preceding) as count
from winsales
order by salesid;
```

```
salesid | qty | count
---------+-----+-----
10001 |  10 |   1
10005 |  30 |   2
10006 |  10 |   3
20001 |  20 |   4
20002 |  20 |   5
30001 |  10 |   6
30003 |  15 |   7
30004 |  20 |   8
30007 |  30 |   9
40001 |  40 |   10
40005 |  10 |   11
(11 rows)
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Nel seguente esempio sono mostrati l'ID vendite, la quantità e il conteggio delle righe non-null dall'inizio della finestra dei dati. (Nella tabella WINSALES, la colonna QTY_SHIPPED contiene alcuni valori NULL.)

```
select salesid, qty, qty_shipped,
count(qty_shipped)
over (order by salesid rows unbounded preceding) as count
from winsales
order by salesid;

salesid | qty | qty_shipped | count
---------+-----+-------------+-------
10001 |  10 |          10 |   1
10005 |  30 |             |   1
10006 |  10 |             |   1
20001 |  20 |          20 |   2
20002 |  20 |          20 |   3
30001 |  10 |          10 |   4
30003 |  15 |             |   4
30004 |  20 |             |   4
30007 |  30 |             |   4
40001 |  40 |             |   4
40005 |  10 |          10 |   5
(11 rows)
```

Funzione finestra CUME_DIST

Calcola la distribuzione cumulativa di un valore all'interno di una finestra o partizione. Supponendo l'ordinamento ascendente, la distribuzione cumulativa è determinata utilizzando questa formula:

```
count of rows with values <= x / count of rows in the window or partition
```

laddove x è uguale al valore nella riga corrente della colonna specificata nella clausola ORDER BY. Il seguente insieme di dati dimostra l'uso di questa formula:

```
Row# Value   Calculation    CUME_DIST
1        2500    (1)/(5)    0.2
2        2600    (2)/(5)    0.4
3        2800    (3)/(5)    0.6
4        2900    (4)/(5)    0.8
5        3100    (5)/(5)    1.0
```

L'intervallo del valore di restituzione è compreso tra 0 e 1, con questi valori compresi.

Sintassi

```
CUME_DIST ()
OVER (
[ PARTITION BY partition_expression ]
[ ORDER BY order_list ]
)
```

Argomenti

OVER

Una clausola che specifica il partizionamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra.

PARTITION BY partition_expression

Facoltativo. Un'espressione che imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

ORDER BY order_list

L'espressione su cui calcolare la distribuzione cumulativa. L'espressione deve avere o un tipo di dati numerici o essere implicitamente convertibile in uno. Se ORDER BY viene omesso, il valore di restituzione è 1 per tutte le righe.

Se ORDER BY non produce un ordinamento univoco, l'ordine delle righe è non deterministico. Per ulteriori informazioni, consultare Ordinamento univoco dei dati per le funzioni finestra.

Tipo restituito

FLOAT8

Esempi

L'esempio seguente calcola la distribuzione cumulativa della quantità per ciascun venditore:

```
select sellerid, qty, cume_dist()
over (partition by sellerid order by qty)
from winsales;

sellerid   qty    cume_dist
------------------------------------------------
1          10.00    0.33
1          10.64    0.67
1          30.37    1
3          10.04    0.25
3          15.15    0.5
3          20.75    0.75
3          30.55    1
2          20.09    0.5
2          20.12    1
4          10.12    0.5
4          40.23    1
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Funzione finestra DENSE_RANK

La funzione finestra DENSE_RANK determina la classificazione di un valore in un gruppo di valori, in base all'espressione ORDER BY nella clausola OVER. Se è presente la clausola PARTITION BY facoltativa, le classificazioni vengono ripristinate per ciascun gruppo di righe. Righe con valori uguali per i criteri di classificazione ricevono la stessa classificazione. La funzione DENSE_RANK differisce da RANK per un aspetto: se due o più righe si legano, non c'è spazio nella sequenza dei valori classificati. Ad esempio, se due righe sono classificate come 1, il livello successivo è 2.

È possibile avere funzioni di classificazione con diverse clausole PARTITION BY e ORDER BY nella stessa query.

Sintassi

```
DENSE_RANK() OVER
(
[ PARTITION BY expr_list ]
[ ORDER BY order_list ]
)
```

Argomenti

( )

La funzione non accetta argomenti, ma le parentesi vuote sono obbligatorie.

OVER

Le clausole finestra per la funzione DENSE_RANK.

PARTITION BY expr_list

(Facoltativo) Una o più espressioni che definiscono la finestra.

ORDER BY order_list

(Facoltativo) L'espressione su cui si basano i valori di classificazione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella. Se ORDER BY viene omesso, il valore di restituzione è 1 per tutte le righe.

Se ORDER BY non produce un ordinamento univoco, l'ordine delle righe è non deterministico. Per ulteriori informazioni, consultare Ordinamento univoco dei dati per le funzioni finestra.

Tipo restituito

`INTEGER`

Esempi

Gli esempi seguenti utilizzano la tabella di esempio per le funzioni delle finestre. Per ulteriori informazioni, consulta Tabella di esempio per gli esempi della funzione finestra.

Nel seguente esempio viene ordinata la tabella in base alla quantità venduta e viene assegnata una classificazione densa e una classificazione regolare a ciascuna riga. I risultati vengono ordinati dopo aver applicato i risultati della funzione finestra.

```
SELECT salesid, qty,
DENSE_RANK() OVER(ORDER BY qty DESC) AS d_rnk,
RANK() OVER(ORDER BY qty DESC) AS rnk
FROM winsales
ORDER BY 2,1;

+---------+-----+-------+-----+
| salesid | qty | d_rnk | rnk |
+---------+-----+-------+-----+
|   10001 |  10 |     5 |   8 |
|   10006 |  10 |     5 |   8 |
|   30001 |  10 |     5 |   8 |
|   40005 |  10 |     5 |   8 |
|   30003 |  15 |     4 |   7 |
|   20001 |  20 |     3 |   4 |
|   20002 |  20 |     3 |   4 |
|   30004 |  20 |     3 |   4 |
|   10005 |  30 |     2 |   2 |
|   30007 |  30 |     2 |   2 |
|   40001 |  40 |     1 |   1 |
+---------+-----+-------+-----+
```

Notare la differenza nelle classificazioni assegnate allo stesso insieme di righe quando le funzioni DENSE_RANK e RANK vengono utilizzate fianco a fianco nella stessa query.

Nel seguente esempio la tabella viene partizionata per sellerid, ciascuna partizione viene ordinata in base alla quantità e viene assegnata una classificazione densa a ciascuna riga. I risultati vengono ordinati dopo aver applicato i risultati della funzione finestra.

```
SELECT salesid, sellerid, qty,
DENSE_RANK() OVER(PARTITION BY sellerid ORDER BY qty DESC) AS d_rnk
FROM winsales
ORDER BY 2,3,1;

+---------+----------+-----+-------+
| salesid | sellerid | qty | d_rnk |
+---------+----------+-----+-------+
|   10001 |        1 |  10 |     2 |
```

```
|   10006 |        1 |  10 |     2 |
|   10005 |        1 |  30 |     1 |
|   20001 |        2 |  20 |     1 |
|   20002 |        2 |  20 |     1 |
|   30001 |        3 |  10 |     4 |
|   30003 |        3 |  15 |     3 |
|   30004 |        3 |  20 |     2 |
|   30007 |        3 |  30 |     1 |
|   40005 |        4 |  10 |     2 |
|   40001 |        4 |  40 |     1 |
+---------+----------+-----+-------+
```

Per utilizzare correttamente l'ultimo esempio, utilizza il comando seguente per inserire una riga nella tabella WINSALES. Questa riga ha lo stesso buyerid, sellerid e qtysold di un'altra riga. Ciò causerà il collegamento di due righe nell'ultimo esempio e quindi mostrerà la differenza tra le funzioni DENSE_RANK e RANK.

```
INSERT INTO winsales VALUES(30009, '2/2/2003', 3, 'b', 20, NULL);
```

Nel seguente esempio la tabella viene partizionata per buyerid e sellerid, ciascuna partizione viene ordinata in base alla quantità e viene assegnata una classificazione densa e una classificazione regolare a ciascuna riga. I risultati vengono ordinati dopo aver applicato la funzione finestra.

```
SELECT salesid, sellerid, qty, buyerid,
DENSE_RANK() OVER(PARTITION BY buyerid, sellerid ORDER BY qty DESC) AS d_rnk,
RANK() OVER (PARTITION BY buyerid, sellerid ORDER BY qty DESC) AS rnk
FROM winsales
ORDER BY rnk;

+---------+----------+-----+---------+-------+-----+
| salesid | sellerid | qty | buyerid | d_rnk | rnk |
+---------+----------+-----+---------+-------+-----+
|   20001 |        2 |  20 | b       |     1 |   1 |
|   30007 |        3 |  30 | c       |     1 |   1 |
|   10006 |        1 |  10 | c       |     1 |   1 |
|   10005 |        1 |  30 | a       |     1 |   1 |
|   20002 |        2 |  20 | c       |     1 |   1 |
|   30009 |        3 |  20 | b       |     1 |   1 |
|   40001 |        4 |  40 | a       |     1 |   1 |
|   30004 |        3 |  20 | b       |     1 |   1 |
|   10001 |        1 |  10 | c       |     1 |   1 |
|   40005 |        4 |  10 | a       |     2 |   2 |
```

```
|   30003 |        3 |  15 | b       |     2 |   3 |
|   30001 |        3 |  10 | b       |     3 |   4 |
+---------+----------+-----+---------+-------+-----+
```

Funzione finestra FIRST_VALUE

Dato un insieme ordinato di righe, FIRST_VALUE restituisce il valore dell'espressione specificata rispetto alla prima riga nel frame della finestra.

Per informazioni sulla selezione dell'ultima riga nel frame, consulta Funzione finestra LAST_VALUE.

Sintassi

```
FIRST_VALUE( expression )[ IGNORE NULLS | RESPECT NULLS ]
OVER (
[ PARTITION BY expr_list ]
[ ORDER BY order_list frame_clause ]
)
```

Argomenti

espressione

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

IGNORE NULLS

Quando questa opzione viene utilizzata con FIRST_VALUE, la funzione restituisce il primo valore nel frame che non è NULL (o NULL se tutti i valori sono NULL).

RESPECT NULLS

Indica che Amazon Redshift dovrebbe includere valori null nella determinazione della riga da utilizzare. RESPECT NULLS è supportato come impostazione predefinita se non si specifica IGNORE NULLS.

OVER

Presenta le clausole finestra per la funzione.

PARTITION BY expr_list

Definisce la finestra per la funzione in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificata nessuna clausola PARTITION BY, ORDER BY ordina l'intera tabella. Se si specifica una clausola ORDER BY, è necessario anche specificare una frame_clause.

I risultati della funzione FIRST_VALUE dipendono dall'ordinamento dei dati. I risultati sono non deterministici nei seguenti casi:

- Quando non è specificata alcuna clausola ORDER BY e una partizione contiene due valori diversi per un'espressione
- Quando l'espressione valuta valori diversi che corrispondono allo stesso valore nell'elenco ORDER BY.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe nel risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipo restituito

Queste funzioni supportano le espressioni che usano tipi di dati primitivi di Amazon Redshift. Il tipo restituito è lo stesso del tipo di dati di expression.

Esempi

Gli esempi seguenti utilizzano la tabella VENUE dai dati di esempio di TICKIT. Per ulteriori informazioni, consulta Database di esempio.

L'esempio seguente restituisce la capacità di posto per ciascuna posizione nella tabella VENUE, con i risultati ordinati in base alla capacità (da alta a bassa). La funzione FIRST_VALUE viene utilizzata per selezionare il nome del luogo corrispondente alla prima riga nel frame: in questo caso, la riga con il numero più alto di posti. I risultati sono partizionati per stato, quindi quando il valore VENUESTATE cambia, viene selezionato un nuovo primo valore. Il frame della finestra è illimitato, quindi lo stesso primo valore è selezionato per ogni riga in ogni partizione.

Per la California, `Qualcomm Stadium` ha il più alto numero di posti (`70561`), quindi questo nome è il primo valore per tutte le righe nella partizione `CA`.

```
select venuestate, venueseats, venuename,
first_value(venuename)
over(partition by venuestate
order by venueseats desc
rows between unbounded preceding and unbounded following)
from (select * from venue where venueseats >0)
order by venuestate;

venuestate | venueseats |           venuename            |         first_value
-----------+------------+--------------------------------
+-----------------------------
CA         |      70561 | Qualcomm Stadium               | Qualcomm Stadium
CA         |      69843 | Monster Park                   | Qualcomm Stadium
CA         |      63026 | McAfee Coliseum                | Qualcomm Stadium
CA         |      56000 | Dodger Stadium                 | Qualcomm Stadium
CA         |      45050 | Angel Stadium of Anaheim       | Qualcomm Stadium
CA         |      42445 | PETCO Park                     | Qualcomm Stadium
CA         |      41503 | AT&T Park                      | Qualcomm Stadium
CA         |      22000 | Shoreline Amphitheatre         | Qualcomm Stadium
CO         |      76125 | INVESCO Field                  | INVESCO Field
CO         |      50445 | Coors Field                    | INVESCO Field
DC         |      41888 | Nationals Park                 | Nationals Park
FL         |      74916 | Dolphin Stadium                | Dolphin Stadium
FL         |      73800 | Jacksonville Municipal Stadium | Dolphin Stadium
FL         |      65647 | Raymond James Stadium          | Dolphin Stadium
FL         |      36048 | Tropicana Field                | Dolphin Stadium
...
```

Il seguente esempio mostra l'uso dell'opzione IGNORE NULLS e fa affidamento sull'aggiunta di una nuova riga nella tabella VENUE:

```
insert into venue values(2000,null,'Stanford','CA',90000);
```

Questa nuova riga contiene un valore NULL per la colonna VENUENAME. Ora ripetere la query FIRST_VALUE mostrata in precedenza in questa sezione:

```
select venuestate, venueseats, venuename,
first_value(venuename)
over(partition by venuestate
order by venueseats desc
rows between unbounded preceding and unbounded following)
from (select * from venue where venueseats >0)
```

```
order by venuestate;

venuestate | venueseats |         venuename          | first_value
-----------+------------+----------------------------+-------------
CA         |      90000 | NULL                       | NULL
CA         |      70561 | Qualcomm Stadium           | NULL
CA         |      69843 | Monster Park               | NULL
...
```

Perché la nuova riga contiene il valore VENUESEATS più alto (90000) e il suo VENUENAME è NULL, la funzione FIRST_VALUE restituisce NULL per la partizione CA. Per ignorare righe come questa nella valutazione della funzione, aggiungere l'opzione IGNORE NULLS all'argomento della funzione:

```
select venuestate, venueseats, venuename,
first_value(venuename) ignore nulls
over(partition by venuestate
order by venueseats desc
rows between unbounded preceding and unbounded following)
from (select * from venue where venuestate='CA')
order by venuestate;

venuestate | venueseats |         venuename          |   first_value
-----------+------------+----------------------------+------------------
CA         |      90000 | NULL                       | Qualcomm Stadium
CA         |      70561 | Qualcomm Stadium           | Qualcomm Stadium
CA         |      69843 | Monster Park               | Qualcomm Stadium
...
```

Funzione finestra LAG

La funzione finestra LAG restituisce i valori per una riga a una data compensazione sopra (prima) la riga corrente nella partizione.

Sintassi

```
LAG (value_expr [, offset ])
[ IGNORE NULLS | RESPECT NULLS ]
OVER ( [ PARTITION BY window_partition ] ORDER BY window_ordering )
```

Argomenti

value_expr

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

offset

Un parametro facoltativo che specifica il numero di righe prima della riga corrente per le quali restituire i valori. La compensazione può essere un integer costante o un'espressione che valuta un integer. Se non viene specificato un offset, Amazon Redshift utilizza 1 come valore predefinito. Una compensazione di `0` indica la riga corrente.

IGNORE NULLS

Una specifica facoltativa che indica che Amazon Redshift dovrebbe saltare i valori null nella determinazione della riga da utilizzare. I valori null sono inclusi se IGNORE NULLS non è elencato.

> **Note**
>
> È possibile utilizzare un'espressione NVL o COALESCE per sostituire i valori null con un altro valore. Per ulteriori informazioni, consultare Funzioni NVL e COALESCE.

RESPECT NULLS

Indica che Amazon Redshift dovrebbe includere valori null nella determinazione della riga da utilizzare. RESPECT NULLS è supportato come impostazione predefinita se non si specifica IGNORE NULLS.

OVER

Specifica il partizionamento e l'ordinamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra.

PARTITION BY window_partition

Un argomento facoltativo che imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

ORDER BY window_ordering

Ordina le righe all'interno di ogni partizione.

La funzione finestra LAG supporta le espressioni che usano uno dei tipi di dati di Amazon Redshift. Il tipo di restituzione è lo stesso del tipo di dati di value_expr.

Esempi

L'esempio seguente mostra la quantità di biglietti venduti all'acquirente con un ID acquirente di 3 e il tempo in cui l'acquirente 3 ha acquistato i biglietti. Per confrontare ogni vendita con la vendita precedente per l'acquirente 3, la query restituisce la quantità venduta per ogni vendita precedente. Poiché non è stato effettuato alcun acquisto prima del 16/01/2008, il primo valore venduto precedentemente è null:

```
select buyerid, saletime, qtysold,
lag(qtysold,1) over (order by buyerid, saletime) as prev_qtysold
from sales where buyerid = 3 order by buyerid, saletime;

buyerid |      saletime       | qtysold | prev_qtysold
---------+---------------------+---------+--------------
3 | 2008-01-16 01:06:09 |       1 |
3 | 2008-01-28 02:10:01 |       1 |            1
3 | 2008-03-12 10:39:53 |       1 |            1
3 | 2008-03-13 02:56:07 |       1 |            1
3 | 2008-03-29 08:21:39 |       2 |            1
3 | 2008-04-27 02:39:01 |       1 |            2
3 | 2008-08-16 07:04:37 |       2 |            1
3 | 2008-08-22 11:45:26 |       2 |            2
3 | 2008-09-12 09:11:25 |       1 |            2
3 | 2008-10-01 06:22:37 |       1 |            1
3 | 2008-10-20 01:55:51 |       2 |            1
3 | 2008-10-28 01:30:40 |       1 |            2
(12 rows)
```

Funzione finestra LAST_VALUE

In un set di righe ordinato, la funzione LAST_VALUE restituisce il valore dell'espressione rispetto all'ultima riga nel frame.

Per informazioni sulla selezione della prima riga nel frame, consulta Funzione finestra FIRST_VALUE.

Sintassi

```
LAST_VALUE( expression )[ IGNORE NULLS | RESPECT NULLS ]
```

```
OVER (
[ PARTITION BY expr_list ]
[ ORDER BY order_list frame_clause ]
)
```

Argomenti

espressione

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

IGNORE NULLS

La funzione restituisce l'ultimo valore nel frame che non è NULL (o NULL se tutti i valori sono NULL).

RESPECT NULLS

Indica che Amazon Redshift dovrebbe includere valori null nella determinazione della riga da utilizzare. RESPECT NULLS è supportato come impostazione predefinita se non si specifica IGNORE NULLS.

OVER

Presenta le clausole finestra per la funzione.

PARTITION BY expr_list

Definisce la finestra per la funzione in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificata nessuna clausola PARTITION BY, ORDER BY ordina l'intera tabella. Se si specifica una clausola ORDER BY, è necessario anche specificare una frame_clause.

I risultati dipendono dall'ordinamento dei dati. I risultati sono non deterministici nei seguenti casi:

- Quando non è specificata alcuna clausola ORDER BY e una partizione contiene due valori diversi per un'espressione
- Quando l'espressione valuta valori diversi che corrispondono allo stesso valore nell'elenco ORDER BY.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della

funzione, includendo o escludendo insieme di righe nel risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipo restituito

Queste funzioni supportano le espressioni che usano tipi di dati primitivi di Amazon Redshift. Il tipo restituito è lo stesso del tipo di dati di expression.

Esempi

Gli esempi seguenti utilizzano la tabella VENUE dai dati di esempio di TICKIT. Per ulteriori informazioni, consulta Database di esempio.

L'esempio seguente restituisce la capacità di posto per ciascuna posizione nella tabella VENUE, con i risultati ordinati in base alla capacità (da alta a bassa). La funzione LAST_VALUE viene utilizzata per selezionare il nome del luogo corrispondente all'ultima riga nel frame: in questo caso, la riga con il numero più basso di posti. I risultati sono partizionati per stato, quindi quando il valore VENUESTATE cambia, viene selezionato un nuovo ultimo valore. Il frame della finestra è illimitato, quindi lo stesso ultimo valore è selezionato per ogni riga in ogni partizione.

Per la California, `Shoreline Amphitheatre` viene restituito per ogni riga nella partizione perché ha il numero più basso di posti (`22000`).

```
select venuestate, venueseats, venuename,
last_value(venuename)
over(partition by venuestate
order by venueseats desc
rows between unbounded preceding and unbounded following)
from (select * from venue where venueseats >0)
order by venuestate;

venuestate | venueseats |          venuename            |          last_value
-----------+------------+-------------------------------
+-----------------------------
CA         |      70561 | Qualcomm Stadium              | Shoreline Amphitheatre
CA         |      69843 | Monster Park                  | Shoreline Amphitheatre
CA         |      63026 | McAfee Coliseum               | Shoreline Amphitheatre
CA         |      56000 | Dodger Stadium                | Shoreline Amphitheatre
CA         |      45050 | Angel Stadium of Anaheim      | Shoreline Amphitheatre
CA         |      42445 | PETCO Park                    | Shoreline Amphitheatre
```

```
CA         |      41503 | AT&T Park                       | Shoreline Amphitheatre
CA         |      22000 | Shoreline Amphitheatre          | Shoreline Amphitheatre
CO         |      76125 | INVESCO Field                   | Coors Field
CO         |      50445 | Coors Field                     | Coors Field
DC         |      41888 | Nationals Park                  | Nationals Park
FL         |      74916 | Dolphin Stadium                 | Tropicana Field
FL         |      73800 | Jacksonville Municipal Stadium | Tropicana Field
FL         |      65647 | Raymond James Stadium           | Tropicana Field
FL         |      36048 | Tropicana Field                 | Tropicana Field
...
```

Funzione finestra LEAD

La funzione finestra LEAD restituisce i valori per una riga a una data compensazione sotto (dopo) la riga corrente nella partizione.

Sintassi

```
LEAD (value_expr [, offset ])
[ IGNORE NULLS | RESPECT NULLS ]
OVER ( [ PARTITION BY window_partition ] ORDER BY window_ordering )
```

Argomenti

value_expr

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

offset

Un parametro facoltativo che specifica il numero di righe sotto la riga corrente per le quali restituire i valori. La compensazione può essere un integer costante o un'espressione che valuta un integer. Se non viene specificato un offset, Amazon Redshift utilizza 1 come valore predefinito. Una compensazione di `0` indica la riga corrente.

IGNORE NULLS

Una specifica facoltativa che indica che Amazon Redshift dovrebbe saltare i valori null nella determinazione della riga da utilizzare. I valori null sono inclusi se IGNORE NULLS non è elencato.

> **Note**
>
> È possibile utilizzare un'espressione NVL o COALESCE per sostituire i valori null con un altro valore. Per ulteriori informazioni, consultare Funzioni NVL e COALESCE.

RESPECT NULLS

Indica che Amazon Redshift dovrebbe includere valori null nella determinazione della riga da utilizzare. RESPECT NULLS è supportato come impostazione predefinita se non si specifica IGNORE NULLS.

OVER

Specifica il partizionamento e l'ordinamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra.

PARTITION BY window_partition

Un argomento facoltativo che imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

ORDER BY window_ordering

Ordina le righe all'interno di ogni partizione.

La funzione finestra LEAD supporta le espressioni che usano uno dei tipi di dati di Amazon Redshift. Il tipo di restituzione è lo stesso del tipo di dati di value_expr.

Esempi

L'esempio seguente fornisce la commissione per gli eventi nella tabella SALES per cui i biglietti sono stati venduti il 1° gennaio 2008 e il 2 gennaio 2008 e la commissione pagata per le vendite dei biglietti per la vendita successiva. L'esempio seguente utilizza il database di esempio TICKIT. Per ulteriori informazioni, consulta Database di esempio.

```
SELECT eventid, commission, saletime, LEAD(commission, 1) over ( ORDER BY saletime ) AS
 next_comm
FROM sales
WHERE saletime BETWEEN '2008-01-09 00:00:00' AND '2008-01-10 12:59:59'
LIMIT 10;

+---------+------------+---------------------+-----------+
```

```
| eventid | commission |      saletime       | next_comm |
+---------+------------+---------------------+-----------+
|    1664 |       13.2 | 2008-01-09 01:00:21 |      69.6 |
|     184 |       69.6 | 2008-01-09 01:00:36 |     116.1 |
|    6870 |      116.1 | 2008-01-09 01:02:37 |      11.1 |
|    3718 |       11.1 | 2008-01-09 01:05:19 |     205.5 |
|    6772 |      205.5 | 2008-01-09 01:14:04 |      38.4 |
|    3074 |       38.4 | 2008-01-09 01:26:50 |     209.4 |
|    5254 |      209.4 | 2008-01-09 01:29:16 |      26.4 |
|    3724 |       26.4 | 2008-01-09 01:40:09 |      57.6 |
|    5303 |       57.6 | 2008-01-09 01:40:21 |      51.6 |
|    3678 |       51.6 | 2008-01-09 01:42:54 |      43.8 |
+---------+------------+---------------------+-----------+
```

Funzione finestra LISTAGG

Per ogni gruppo in una query, la funzione finestra LISTAGG ordina le righe di quel gruppo in base all'espressione ORDER BY, quindi concatena i valori in un'unica stringa.

LISTAGG è una funzione solo del nodo di calcolo. La funzione restituisce un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift. Per ulteriori informazioni, consulta Query sulle tabelle di catalogo.

Sintassi

```
LISTAGG( [DISTINCT] expression [, 'delimiter' ] )
[ WITHIN GROUP (ORDER BY order_list) ]
OVER ( [PARTITION BY partition_expression] )
```

Argomenti

DISTINCT

(Facoltativo) Una clausola che elimina i valori duplicati dall'espressione specificata prima della concatenazione. Gli spazi finali vengono ignorati, quindi le stringhe `'a'` e `'a '` vengono trattate come duplicati. LISTAGG usa il primo valore incontrato. Per ulteriori informazioni, consultare Significato degli spazi finali.

aggregate_expression

Qualsiasi espressione valida (come il nome di una colonna) che fornisce i valori da aggregare. I valori NULL e le stringhe vuote vengono ignorati.

delimiter

(Facoltativo) La costante di stringa separerà i valori concatenati. Il valore predefinito è NULL.

WITHIN GROUP (ORDER BY order_list)

(Facoltativo) Una clausola che specifica l'ordinamento dei valori aggregati. Deterministico solo se ORDER BY fornisce un ordinamento univoco. L'impostazione predefinita è quella di aggregare tutte le righe e restituire un valore singolo.

OVER

Una clausola che specifica il partizionamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra o dell'ordinamento della finestra.

PARTITION BY partition_expression

(Facoltativo) Imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

Valori restituiti

VARCHAR(MAX). Se l'insieme dei risultati è maggiore della dimensione massima di VARCHAR (64K - 1, oppure 65535), allora LISTAGG restituisce il seguente errore:

```
Invalid operation: Result size exceeds LISTAGG limit
```

Esempi

Gli esempi seguenti usano la tabella WINSALES. Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

L'esempio seguente restituisce un elenco di ID venditore, ordinati per ID venditore.

```
select listagg(sellerid)
within group (order by sellerid)
over() from winsales;

  listagg
------------
 11122333344
...
...
 11122333344
 11122333344
```

```
   (11 rows)
```

L'esempio seguente restituisce un elenco di ID venditore per acquirente B, ordinati in base alla data.

```
select listagg(sellerid)
within group (order by dateid)
over () as seller
from winsales
where buyerid = 'b' ;

  seller
---------
    3233
    3233
    3233
    3233
```

Il seguente esempio restituisce un elenco separato dalla virgola di date di vendita per l'acquirente B.

```
select listagg(dateid,',')
within group (order by sellerid desc,salesid asc)
over () as dates
from winsales
where buyerid  = 'b';

             dates
----------------------------------------
2003-08-02,2004-04-18,2004-04-18,2004-02-12
2003-08-02,2004-04-18,2004-04-18,2004-02-12
2003-08-02,2004-04-18,2004-04-18,2004-02-12
2003-08-02,2004-04-18,2004-04-18,2004-02-12
```

Il seguente esempio utilizza DISTINCT per restituire un elenco di date di vendita univoche per l'acquirente B.

```
select listagg(distinct dateid,',')
within group (order by sellerid desc,salesid asc)
over () as dates
from winsales
where buyerid  = 'b';

           dates
```

```
-------------------------------
2003-08-02,2004-04-18,2004-02-12
2003-08-02,2004-04-18,2004-02-12
2003-08-02,2004-04-18,2004-02-12
2003-08-02,2004-04-18,2004-02-12
```

Il seguente esempio restituisce un elenco separato dalla virgola di ID di vendita per ciascun ID acquirente.

```
select buyerid,
listagg(salesid,',')
within group (order by salesid)
over (partition by buyerid) as sales_id
from winsales
order by buyerid;

+---------+------------------------+
| buyerid |        sales_id        |
+---------+------------------------+
| a       | 10005,40001,40005      |
| a       | 10005,40001,40005      |
| a       | 10005,40001,40005      |
| b       | 20001,30001,30003,30004 |
| b       | 20001,30001,30003,30004 |
| b       | 20001,30001,30003,30004 |
| b       | 20001,30001,30003,30004 |
| c       | 10001,10006,20002,30007 |
| c       | 10001,10006,20002,30007 |
| c       | 10001,10006,20002,30007 |
| c       | 10001,10006,20002,30007 |
+---------+------------------------+
```

Funzione finestra MAX

La funzione finestra MAX restituisce il massimo dei valori dell'espressione di input. La funzione MAX funziona con i valori numerici e ignora i valori NULL.

Sintassi

```
MAX ( [ ALL ] expression ) OVER
(
[ PARTITION BY expr_list ]
```

```
[ ORDER BY order_list frame_clause ]
)
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

ALL

Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione. ALL è il valore predefinito. DISTINCT non è supportato.

OVER

Una clausola che specifica le clausole finestra per le funzioni di aggregazione. La clausola OVER distingue le funzioni di aggregazione delle finestre dalle normali funzioni di aggregazione dell'insieme.

PARTITION BY expr_list

Definisce la finestra per la funzione MAX in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe all'interno del risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipi di dati

Accetta qualsiasi tipo di dati come input. Restituisce lo stesso tipo di dati come espressione.

Esempi

Nel seguente esempio sono riportati l'ID vendite, la quantità e la quantità massima dall'inizio della finestra dei dati:

```
select salesid, qty,
max(qty) over (order by salesid rows unbounded preceding) as max
from winsales
order by salesid;

salesid | qty | max
---------+-----+-----
10001 |  10 |  10
10005 |  30 |  30
10006 |  10 |  30
20001 |  20 |  30
20002 |  20 |  30
30001 |  10 |  30
30003 |  15 |  30
30004 |  20 |  30
30007 |  30 |  30
40001 |  40 |  40
40005 |  10 |  40
(11 rows)
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Nel seguente esempio sono mostrati l'ID di vendita, la quantità e la quantità massima in un frame limitato:

```
select salesid, qty,
max(qty) over (order by salesid rows between 2 preceding and 1 preceding) as max
from winsales
order by salesid;

salesid | qty | max
---------+-----+-----
10001 |  10 |
10005 |  30 |  10
10006 |  10 |  30
20001 |  20 |  30
20002 |  20 |  20
30001 |  10 |  20
30003 |  15 |  20
30004 |  20 |  15
30007 |  30 |  20
40001 |  40 |  30
```

```
40005 |  10 |  40
(11 rows)
```

Funzione finestra MEDIAN

Calcola il valore mediano per l'intervallo di valori in una finestra o partizione. I valori NULL nell'intervallo vengono ignorati.

MEDIAN è una funzione di distribuzione inversa che presuppone un modello di distribuzione continua.

MEDIAN è una funzione solo del nodo di calcolo. La funzione restituisce un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift.

Sintassi

```
MEDIAN ( median_expression )
OVER ( [ PARTITION BY partition_expression ] )
```

Argomenti

median_expression

Un'espressione, come ad esempio un nome di colonna, che fornisce i valori per cui determinare il valore medio. L'espressione deve avere o un tipo di dati numerici o datetime oppure essere implicitamente convertibile in uno.

OVER

Una clausola che specifica il partizionamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra o dell'ordinamento della finestra.

PARTITION BY partition_expression

Facoltativo. Un'espressione che imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

Tipi di dati

Il tipo di ritorno è determinato dal tipo di dati di median_expression. La seguente tabella mostra il tipo di restituzione per ciascun median_expression tipo di dati.

Tipo di input	Tipo restituito
INT2, INT4, INT8, NUMERIC, DECIMAL	DECIMAL
FLOAT, DOUBLE	DOUBLE
DATE	DATE

Note per l'utilizzo

Se l'argomento median_expression è un tipo di dati DECIMAL definito con la precisione massima di 38 cifre, è possibile che MEDIAN restituirà un risultato inaccurato o un errore. Se il valore di ritorno della funzione MEDIAN supera le 38 cifre, il risultato viene troncato per adattarsi, il che causa una perdita della precisione. Se, durante l'interpolazione, un risultato intermedio supera la precisione massima, si verifica un'eccedenza numerica e la funzione restituisce un errore. Per evitare queste condizioni, consigliamo di utilizzare un tipo di dati con una precisione inferiore o di assegnare l'argomento median_expression a una precisione inferiore.

Ad esempio, una funzione SUM con un argomento DECIMAL restituisce una precisione predefinita di 38 cifre. Il ridimensionamento del risultato coincide con il ridimensionamento dell'argomento. Quindi, ad esempio, un SUM di una colonna DECIMAL(5,2) restituisce un tipo di dati DECIMAL(38,2).

L'esempio seguente utilizza una funzione SUM nell'argomento median_expression di una funzione MEDIAN. Il tipo di dati della colonna PRICEPAID è DECIMAL (8,2), quindi la funzione SUM restituisce DECIMAL (38,2).

```
select salesid, sum(pricepaid), median(sum(pricepaid))
over() from sales where salesid < 10 group by salesid;
```

Per evitare una perdita potenziale di precisione o un errore di sovraccarico, assegnare il risultato a un tipo di dati DECIMAL con una precisione inferiore, come mostra il seguente esempio.

```
select salesid, sum(pricepaid), median(sum(pricepaid)::decimal(30,2))
over() from sales where salesid < 10 group by salesid;
```

Esempi

L'esempio seguente calcola la quantità media di vendite per ciascun venditore:

```
select sellerid, qty, median(qty)
over (partition by sellerid)
from winsales
order by sellerid;


sellerid qty median
---------------------------
1  10 10.0
1  10 10.0
1  30 10.0
2  20 20.0
2  20 20.0
3  10 17.5
3  15 17.5
3  20 17.5
3  30 17.5
4  10 25.0
4  40 25.0
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Funzione finestra MIN

La funzione finestra MIN restituisce il minimo dei valori dell'espressione di input. La funzione MIN funziona con i valori numerici e ignora i valori NULL.

Sintassi

```
MIN ( [ ALL ] expression ) OVER
(
[ PARTITION BY expr_list ]
[ ORDER BY order_list frame_clause ]
)
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

ALL

Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione. ALL è il valore predefinito. DISTINCT non è supportato.

OVER

Specifica le clausole finestra per le funzioni di aggregazione. La clausola OVER distingue le funzioni di aggregazione delle finestre dalle normali funzioni di aggregazione dell'insieme.

PARTITION BY expr_list

Definisce la finestra per la funzione MIN in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe all'interno del risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipi di dati

Accetta qualsiasi tipo di dati come input. Restituisce lo stesso tipo di dati come espressione.

Esempi

Nel seguente esempio sono riportati l'ID vendite, la quantità e la quantità minima dall'inizio della finestra dei dati:

```
select salesid, qty,
min(qty) over
(order by salesid rows unbounded preceding)
from winsales
order by salesid;

salesid | qty | min
---------+-----+-----
10001 |  10 |  10
```

```
10005 |  30 |  10
10006 |  10 |  10
20001 |  20 |  10
20002 |  20 |  10
30001 |  10 |  10
30003 |  15 |  10
30004 |  20 |  10
30007 |  30 |  10
40001 |  40 |  10
40005 |  10 |  10
(11 rows)
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Nel seguente esempio sono mostrati l'ID di vendita, la quantità e la quantità minima in un frame limitato:

```
select salesid, qty,
min(qty) over
(order by salesid rows between 2 preceding and 1 preceding) as min
from winsales
order by salesid;

salesid | qty | min
---------+-----+-----
10001 |  10 |
10005 |  30 |  10
10006 |  10 |  10
20001 |  20 |  10
20002 |  20 |  10
30001 |  10 |  20
30003 |  15 |  10
30004 |  20 |  10
30007 |  30 |  15
40001 |  40 |  20
40005 |  10 |  30
(11 rows)
```

Funzione finestra NTH_VALUE

La funzione finestra di NTH_VALUE restituisce il valore dell'espressione della riga specificata del frame della finestra relativo alla prima riga della finestra.

Sintassi

```
NTH_VALUE (expr, offset)
[ IGNORE NULLS | RESPECT NULLS ]
OVER
( [ PARTITION BY window_partition ]
[ ORDER BY window_ordering
                        frame_clause ] )
```

Argomenti

expr

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

offset

Determina il numero di riga relativo alla prima riga nella finestra per la quale restituire l'espressione. La compensazione può essere una costante o un'espressione e deve essere un integer positivo maggiore di 0.

IGNORE NULLS

Una specifica facoltativa che indica che Amazon Redshift dovrebbe saltare i valori null nella determinazione della riga da utilizzare. I valori null sono inclusi se IGNORE NULLS non è elencato.

RESPECT NULLS

Indica che Amazon Redshift dovrebbe includere valori null nella determinazione della riga da utilizzare. RESPECT NULLS è supportato come impostazione predefinita se non si specifica IGNORE NULLS.

OVER

Specifica il partizionamento e l'ordinamento della finestra e il frame della finestra.

PARTITION BY window_partition

Imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

ORDER BY window_ordering

Ordina le righe all'interno di ogni partizione. Se viene omesso ORDER BY, il frame predefinito è composto da tutte le righe nella partizione.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe nel risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

La funzione finestra NTH_VALUE supporta le espressioni che usano uno dei tipi di dati di Amazon Redshift. Il tipo di restituzione è lo stesso del tipo di dati di expr.

Esempi

L'esempio seguente mostra il numero di posti nel terzo luogo più grande in California, Florida e New York rispetto al numero di posti negli altri luoghi in quegli stati:

```
select venuestate, venuename, venueseats,
nth_value(venueseats, 3)
ignore nulls
over(partition by venuestate order by venueseats desc
rows between unbounded preceding and unbounded following)
as third_most_seats
from (select * from venue where venueseats > 0 and
venuestate in('CA', 'FL', 'NY'))
order by venuestate;

venuestate |           venuename            | venueseats | third_most_seats
-----------+--------------------------------+------------+------------------
CA         | Qualcomm Stadium               |      70561 |            63026
CA         | Monster Park                   |      69843 |            63026
CA         | McAfee Coliseum                |      63026 |            63026
CA         | Dodger Stadium                 |      56000 |            63026
CA         | Angel Stadium of Anaheim       |      45050 |            63026
CA         | PETCO Park                     |      42445 |            63026
CA         | AT&T Park                      |      41503 |            63026
CA         | Shoreline Amphitheatre         |      22000 |            63026
FL         | Dolphin Stadium                |      74916 |            65647
FL         | Jacksonville Municipal Stadium |      73800 |            65647
FL         | Raymond James Stadium          |      65647 |            65647
FL         | Tropicana Field                |      36048 |            65647
NY         | Ralph Wilson Stadium           |      73967 |            20000
NY         | Yankee Stadium                 |      52325 |            20000
```

```
NY         | Madison Square Garden          |      20000 |            20000
(15 rows)
```

Funzione finestra NTILE

La funzione finestra NTILE divide le righe ordinate nella partizione nel numero specificato di gruppi classificati di dimensioni uguali possibili e restituisce il gruppo in cui cade una determinata riga.

Sintassi

```
NTILE (expr)
OVER (
[ PARTITION BY expression_list ]
[ ORDER BY order_list ]
)
```

Argomenti

expr

Il numero di classificazione si raggruppa e deve avere come risultato un valore intero positivo (maggiore di 0) per ogni partizione. L'argomento expr non deve essere reso null.

OVER

Una clausola che specifica il partizionamento e l'ordinamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra.

PARTITION BY window_partition

Facoltativo. L'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

ORDER BY window_ordering

Facoltativo. Un'espressione che ordina le righe all'interno di ogni partizione. Se viene omessa la clausola ORDER BY, il comportamento di classificazione è lo stesso.

Se ORDER BY non produce un ordinamento univoco, l'ordine delle righe non è deterministico. Per ulteriori informazioni, consultare Ordinamento univoco dei dati per le funzioni finestra.

Tipo restituito

BIGINT

Esempi

Il seguente esempio classifica in quattro gruppi di classificazione il prezzo pagato per i biglietti di Amleto il 26 agosto 2008. Il risultato è di 17 file, divise in modo quasi uniforme tra le classificazioni da 1 a 4:

```
select eventname, caldate, pricepaid, ntile(4)
over(order by pricepaid desc) from sales, event, date
where sales.eventid=event.eventid and event.dateid=date.dateid and eventname='Hamlet'
and caldate='2008-08-26'
order by 4;

eventname |  caldate   | pricepaid | ntile
----------+------------+-----------+-------
Hamlet    | 2008-08-26 |   1883.00 |     1
Hamlet    | 2008-08-26 |   1065.00 |     1
Hamlet    | 2008-08-26 |    589.00 |     1
Hamlet    | 2008-08-26 |    530.00 |     1
Hamlet    | 2008-08-26 |    472.00 |     1
Hamlet    | 2008-08-26 |    460.00 |     2
Hamlet    | 2008-08-26 |    355.00 |     2
Hamlet    | 2008-08-26 |    334.00 |     2
Hamlet    | 2008-08-26 |    296.00 |     2
Hamlet    | 2008-08-26 |    230.00 |     3
Hamlet    | 2008-08-26 |    216.00 |     3
Hamlet    | 2008-08-26 |    212.00 |     3
Hamlet    | 2008-08-26 |    106.00 |     3
Hamlet    | 2008-08-26 |    100.00 |     4
Hamlet    | 2008-08-26 |     94.00 |     4
Hamlet    | 2008-08-26 |     53.00 |     4
Hamlet    | 2008-08-26 |     25.00 |     4
(17 rows)
```

Funzione finestra PERCENT_RANK

Calcola la classificazione percentuale di una data riga. La classificazione percentuale è determinata utilizzando questa formula:

```
(x - 1) / (the number of rows in the window or partition - 1)
```

laddove x è la classificazione della riga corrente. Il seguente insieme di dati dimostra l'uso di questa formula:

```
Row# Value Rank Calculation PERCENT_RANK
1 15 1 (1-1)/(7-1) 0.0000
2 20 2 (2-1)/(7-1) 0.1666
3 20 2 (2-1)/(7-1) 0.1666
4 20 2 (2-1)/(7-1) 0.1666
5 30 5 (5-1)/(7-1) 0.6666
6 30 5 (5-1)/(7-1) 0.6666
7 40 7 (7-1)/(7-1) 1.0000
```

L'intervallo del valore di restituzione è compreso tra 0 e 1, con questi valori compresi. La prima riga in qualsiasi insieme ha un PERCENT_RANK di 0.

Sintassi

```
PERCENT_RANK ()
OVER (
[ PARTITION BY partition_expression ]
[ ORDER BY order_list ]
)
```

Argomenti

( )

La funzione non accetta argomenti, ma le parentesi vuote sono obbligatorie.

OVER

Una clausola che specifica il partizionamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra.

PARTITION BY partition_expression

Facoltativo. Un'espressione che imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

ORDER BY order_list

Facoltativo. L'espressione su cui calcolare la classificazione percentuale. L'espressione deve avere o un tipo di dati numerici o essere implicitamente convertibile in uno. Se ORDER BY viene omesso, il valore di restituzione è 0 per tutte le righe.

Se ORDER BY non produce un ordinamento univoco, l'ordine delle righe non è deterministico. Per ulteriori informazioni, consultare Ordinamento univoco dei dati per le funzioni finestra.

Tipo restituito

FLOAT8

Esempi

L'esempio seguente calcola la classificazione in percentuale delle quantità di vendita per ciascun venditore:

```
select sellerid, qty, percent_rank()
over (partition by sellerid order by qty)
from winsales;

sellerid qty  percent_rank
---------------------------------------
1  10.00  0.0
1  10.64  0.5
1  30.37  1.0
3  10.04  0.0
3  15.15  0.33
3  20.75  0.67
3  30.55  1.0
2  20.09  0.0
2  20.12  1.0
4  10.12  0.0
4  40.23  1.0
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Funzione finestra PERCENTILE_CONT

PERCENTILE_CONT è una funzione di distribuzione inversa che presuppone un modello di distribuzione continua. Prende un valore percentile e una specifica di ordinamento e restituisce un valore interpolato che rientrerebbe nel valore percentile dato rispetto alla specifica di ordinamento.

PERCENTILE_CONT calcola un'interpolazione lineare tra i valori dopo averli ordinati. Usando il valore percentile `(P)` e il numero di righe non null `(N)` nel gruppo di aggregazione, la funzione calcola il numero di righe dopo aver ordinato le righe secondo la specifica di ordinamento. Questo numero di riga `(RN)` è calcolato secondo la formula `RN = (1+ (P*(N-1))`. Il risultato finale della funzione di aggregazione è calcolato mediante interpolazione lineare tra i valori delle righe ai numeri di riga `CRN = CEILING(RN)` e `FRN = FLOOR(RN)`.

Il risultato finale sarà il seguente.

Se `(CRN = FRN = RN)` allora il risultato è `(value of expression from row at RN)`

Altrimenti il risultato è il seguente:

`(CRN - RN) * (value of expression for row at FRN) + (RN - FRN) * (value of expression for row at CRN)`.

È possibile specificare solo la clausola PARTITION nella clausola OVER. Se viene specificata la PARTITION, per ogni riga, PERCENTILE_CONT restituisce il valore che rientrerebbe nel percentile specificato tra un insieme di valori all'interno di una determinata partizione.

PERCENTILE_CONT è una funzione solo del nodo di calcolo. La funzione restituisce un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift.

Sintassi

```
PERCENTILE_CONT ( percentile )
WITHIN GROUP (ORDER BY expr)
OVER (  [ PARTITION BY expr_list ]  )
```

Argomenti

percentile

Costante numerica compresa tra 0 e 1. I valori null vengono ignorati nel calcolo.

WITHIN GROUP (ORDER BY expr)

Specifica i valori numerici o di data/ora per ordinare e calcolare il percentile.

OVER

Specifica il partizionamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra o dell'ordinamento della finestra.

PARTITION BY expr

Argomento facoltativo che imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

Valori restituiti

Il tipo di restituzione è determinato dal tipo di dati dell'espressione ORDER BY nella clausola WITHIN GROUP. La seguente tabella mostra il tipo di restituzione per ciascun tipo di dati dell'espressione ORDER BY.

Tipo di input	Tipo restituito
INT2, INT4, INT8, NUMERIC, DECIMAL	DECIMAL
FLOAT, DOUBLE	DOUBLE
DATE	DATE
TIMESTAMP	TIMESTAMP

Note per l'utilizzo

Se l'espressione ORDER BY è un tipo di dati DECIMAL definito con la precisione massima di 38 cifre, è possibile che PERCENTILE_CONT restituirà un risultato inaccurato o un errore. Se il valore di ritorno della funzione PERCENTILE_CONT supera le 38 cifre, il risultato viene troncato per adattarsi, il che causa una perdita della precisione. Se, durante l'interpolazione, un risultato intermedio supera la precisione massima, si verifica un'eccedenza numerica e la funzione restituisce un errore. Per evitare queste condizioni, consigliamo di utilizzare un tipo di dati con una precisione inferiore o di assegnare l'espressione ORDER BY a una precisione inferiore.

Ad esempio, una funzione SUM con un argomento DECIMAL restituisce una precisione predefinita di 38 cifre. Il ridimensionamento del risultato coincide con il ridimensionamento dell'argomento. Quindi, ad esempio, un SUM di una colonna DECIMAL(5,2) restituisce un tipo di dati DECIMAL(38,2).

L'esempio seguente utilizza una funzione SUM nella clausola ORDER BY di una funzione PERCENTILE_CONT. Il tipo di dati della colonna PRICEPAID è DECIMAL (8,2), quindi la funzione SUM restituisce DECIMAL (38,2).

```
select salesid, sum(pricepaid), percentile_cont(0.6)
within group (order by sum(pricepaid) desc) over()
from sales where salesid < 10 group by salesid;
```

Per evitare una perdita potenziale di precisione o un errore di sovraccarico, assegnare il risultato a un tipo di dati DECIMAL con una precisione inferiore, come mostra il seguente esempio.

```
select salesid, sum(pricepaid), percentile_cont(0.6)
within group (order by sum(pricepaid)::decimal(30,2) desc) over()
from sales where salesid < 10 group by salesid;
```

Esempi

Gli esempi seguenti usano la tabella WINSALES. Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

```
select sellerid, qty, percentile_cont(0.5)
within group (order by qty)
over() as median from winsales;

 sellerid | qty | median
----------+-----+--------
        1 |  10 |   20.0
        1 |  10 |   20.0
        3 |  10 |   20.0
        4 |  10 |   20.0
        3 |  15 |   20.0
        2 |  20 |   20.0
        3 |  20 |   20.0
        2 |  20 |   20.0
        3 |  30 |   20.0
        1 |  30 |   20.0
        4 |  40 |   20.0
(11 rows)
```

```
select sellerid, qty, percentile_cont(0.5)
within group (order by qty)
over(partition by sellerid) as median from winsales;

 sellerid | qty | median
----------+-----+--------
        2 |  20 |   20.0
        2 |  20 |   20.0
        4 |  10 |   25.0
        4 |  40 |   25.0
        1 |  10 |   10.0
```

```
        1 |  10 |   10.0
        1 |  30 |   10.0
        3 |  10 |   17.5
        3 |  15 |   17.5
        3 |  20 |   17.5
        3 |  30 |   17.5
(11 rows)
```

L'esempio seguente calcola le PERCENTILE_CONT e PERCENTILE_DISC delle vendite dei biglietti per i venditori nello stato di Washington.

```
SELECT sellerid, state, sum(qtysold*pricepaid) sales,
percentile_cont(0.6) within group (order by sum(qtysold*pricepaid::decimal(14,2) )
 desc) over(),
percentile_disc(0.6) within group (order by sum(qtysold*pricepaid::decimal(14,2) )
 desc) over()
from sales s, users u
where s.sellerid = u.userid and state = 'WA' and sellerid < 1000
group by sellerid, state;

 sellerid | state |  sales  | percentile_cont | percentile_disc
----------+-------+---------+-----------------+-----------------
      127 | WA    | 6076.00 |         2044.20 |         1531.00
      787 | WA    | 6035.00 |         2044.20 |         1531.00
      381 | WA    | 5881.00 |         2044.20 |         1531.00
      777 | WA    | 2814.00 |         2044.20 |         1531.00
       33 | WA    | 1531.00 |         2044.20 |         1531.00
      800 | WA    | 1476.00 |         2044.20 |         1531.00
        1 | WA    | 1177.00 |         2044.20 |         1531.00
(7 rows)
```

Funzione finestra PERCENTILE_DISC

PERCENTILE_DISC è una funzione di distribuzione inversa che presuppone un modello di distribuzione discreta. Prende un valore percentile e una specifica di ordinamento e restituisce un elemento dall'insieme specificato.

Per un dato valore percentile P, PERCENTILE_DISC ordina i valori dell'espressione nella clausola ORDER BY e restituisce il valore con il valore di distribuzione cumulativo più piccolo (rispetto alla stessa specifica di ordinamento) che è maggiore o uguale a P.

È possibile specificare solo la clausola PARTITION nella clausola OVER.

PERCENTILE_DISC è una funzione solo del nodo di calcolo. La funzione restituisce un errore se la query non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift.

Sintassi

```
PERCENTILE_DISC ( percentile )
WITHIN GROUP (ORDER BY expr)
OVER (  [ PARTITION BY expr_list ]  )
```

Argomenti

percentile

Costante numerica compresa tra 0 e 1. I valori null vengono ignorati nel calcolo.

WITHIN GROUP (ORDER BY expr)

Specifica i valori numerici o di data/ora per ordinare e calcolare il percentile.

OVER

Specifica il partizionamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra o dell'ordinamento della finestra.

PARTITION BY expr

Argomento facoltativo che imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

Valori restituiti

Lo stesso tipo di dati dell'espressione ORDER BY nella clausola WITHIN GROUP.

Esempi

Gli esempi seguenti usano la tabella WINSALES. Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

```
SELECT sellerid, qty, PERCENTILE_DISC(0.5)
WITHIN GROUP (ORDER BY qty)
OVER() AS MEDIAN FROM winsales;
```

```
+----------+-----+--------+
| sellerid | qty | median |
+----------+-----+--------+
| 3        | 10  | 20     |
| 1        | 10  | 20     |
| 1        | 10  | 20     |
| 4        | 10  | 20     |
| 3        | 15  | 20     |
| 2        | 20  | 20     |
| 2        | 20  | 20     |
| 3        | 20  | 20     |
| 1        | 30  | 20     |
| 3        | 30  | 20     |
| 4        | 40  | 20     |
+----------+-----+--------+

SELECT sellerid, qty, PERCENTILE_DISC(0.5)
WITHIN GROUP (ORDER BY qty)
OVER(PARTITION BY sellerid) AS MEDIAN FROM winsales;

+----------+-----+--------+
| sellerid | qty | median |
+----------+-----+--------+
| 4        | 10  | 10     |
| 4        | 40  | 10     |
| 3        | 10  | 15     |
| 3        | 15  | 15     |
| 3        | 20  | 15     |
| 3        | 30  | 15     |
| 2        | 20  | 20     |
| 2        | 20  | 20     |
| 1        | 10  | 10     |
| 1        | 10  | 10     |
| 1        | 30  | 10     |
+----------+-----+--------+
```

Per trovare PERCENTILE_DISC (0.25) e PERCENTILE_DISC (0.75) per la quantità quando viene partizionata in base all'ID del venditore, usa i seguenti esempi.

```
SELECT sellerid, qty, PERCENTILE_DISC(0.25)
WITHIN GROUP (ORDER BY qty)
OVER(PARTITION BY sellerid) AS quartile1 FROM winsales;
```

```
+----------+-----+-----------+
| sellerid | qty | quartile1 |
+----------+-----+-----------+
| 4        | 10  | 10        |
| 4        | 40  | 10        |
| 2        | 20  | 20        |
| 2        | 20  | 20        |
| 3        | 10  | 10        |
| 3        | 15  | 10        |
| 3        | 20  | 10        |
| 3        | 30  | 10        |
| 1        | 10  | 10        |
| 1        | 10  | 10        |
| 1        | 30  | 10        |
+----------+-----+-----------+

SELECT sellerid, qty, PERCENTILE_DISC(0.75)
WITHIN GROUP (ORDER BY qty)
OVER(PARTITION BY sellerid) AS quartile3 FROM winsales;

+----------+-----+-----------+
| sellerid | qty | quartile3 |
+----------+-----+-----------+
| 3        | 10  | 20        |
| 3        | 15  | 20        |
| 3        | 20  | 20        |
| 3        | 30  | 20        |
| 4        | 10  | 40        |
| 4        | 40  | 40        |
| 2        | 20  | 20        |
| 2        | 20  | 20        |
| 1        | 10  | 30        |
| 1        | 10  | 30        |
| 1        | 30  | 30        |
+----------+-----+-----------+
```

Funzione finestra RANK

La funzione finestra RANK determina la classificazione di un valore in un gruppo di valori, in base all'espressione ORDER BY nella clausola OVER. Se è presente la clausola PARTITION BY facoltativa, le classificazioni vengono ripristinate per ciascun gruppo di righe. Righe con valori uguali per i criteri di classificazione ricevono la stessa classificazione. Amazon Redshift aggiunge il numero di righe vincolate alla classificazione vincolata per calcolare la classificazione successiva pertanto le

classificazioni potrebbero non essere numeri consecutivi. Ad esempio, se due righe sono classificate come 1, il livello successivo è 3.

RANK è diverso dalla Funzione finestra DENSE_RANK per un aspetto: Per DENSE_RANK, se due o più righe si legano, non c'è spazio nella sequenza dei valori classificati. Ad esempio, se due righe sono classificate come 1, il livello successivo è 2.

È possibile avere funzioni di classificazione con diverse clausole PARTITION BY e ORDER BY nella stessa query.

Sintassi

```
RANK () OVER
(
[ PARTITION BY expr_list ]
[ ORDER BY order_list ]
)
```

Argomenti

( )

La funzione non accetta argomenti, ma le parentesi vuote sono obbligatorie.

OVER

Le clausole finestra per la funzione RANK.

PARTITION BY expr_list

Facoltativo. Una o più espressioni che definiscono la finestra.

ORDER BY order_list

Facoltativo. Definisce le colonne su cui si basano i valori di classificazione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella. Se ORDER BY viene omesso, il valore di restituzione è 1 per tutte le righe.

Se ORDER BY non produce un ordinamento univoco, l'ordine delle righe non è deterministico. Per ulteriori informazioni, consultare Ordinamento univoco dei dati per le funzioni finestra.

Tipo restituito

INTEGER

Esempi

Nel seguente esempio la tabella viene ordinata in base alla quantità venduta (in ordine crescente per impostazione predefinita) e viene assegnata una classificazione a ciascuna riga. Un valore di classificazione pari a 1 è il valore di classificazione più alto. I risultati vengono ordinati dopo aver applicato i risultati della funzione finestra:

```
select salesid, qty,
rank() over (order by qty) as rnk
from winsales
order by 2,1;

salesid | qty | rnk
--------+-----+-----
10001 |  10 |  1
10006 |  10 |  1
30001 |  10 |  1
40005 |  10 |  1
30003 |  15 |  5
20001 |  20 |  6
20002 |  20 |  6
30004 |  20 |  6
10005 |  30 |  9
30007 |  30 |  9
40001 |  40 |  11
(11 rows)
```

Da notare che la clausola ORDER BY esterna in questo esempio comprende le colonne 2 e 1 per assicurare che Amazon Redshift restituisca risultati ordinati in modo coerente ogni volta che viene eseguita questa query. Ad esempio, le righe con ID vendite 10001 e 10006 hanno valori QTY e RNK identici. Ordinare il risultato finale impostato in base alla colonna 1 assicura che la riga 10001 cada sempre prima di 10006. Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Nel seguente esempio, l'ordinamento è invertito per la funzione finestra (`order by qty desc`). Ora il valore di classificazione più alto si applica al valore QTY più alto.

```
select salesid, qty,
rank() over (order by qty desc) as rank
from winsales
order by 2,1;
```

```
 salesid | qty | rank
---------+-----+-----
   10001 |  10 |   8
   10006 |  10 |   8
   30001 |  10 |   8
   40005 |  10 |   8
   30003 |  15 |   7
   20001 |  20 |   4
   20002 |  20 |   4
   30004 |  20 |   4
   10005 |  30 |   2
   30007 |  30 |   2
   40001 |  40 |   1
(11 rows)
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Nel seguente esempio la tabella viene partizionata per SELLERID, ciascuna partizione viene ordinata in base alla quantità (in ordine decrescente) e viene assegnata una classificazione a ciascuna riga. I risultati vengono ordinati dopo aver applicato i risultati della funzione finestra.

```
select salesid, sellerid, qty, rank() over
(partition by sellerid
order by qty desc) as rank
from winsales
order by 2,3,1;

salesid | sellerid | qty | rank
--------+----------+-----+-----
  10001 |        1 |  10 |  2
  10006 |        1 |  10 |  2
  10005 |        1 |  30 |  1
  20001 |        2 |  20 |  1
  20002 |        2 |  20 |  1
  30001 |        3 |  10 |  4
  30003 |        3 |  15 |  3
  30004 |        3 |  20 |  2
  30007 |        3 |  30 |  1
  40005 |        4 |  10 |  2
  40001 |        4 |  40 |  1
(11 rows)
```

Funzione finestra RATIO_TO_REPORT

Calcola il rapporto di un valore per la somma dei valori in una finestra o partizione. Il rapporto con il valore del report viene determinato utilizzando la formula:

`value of` ratio_expression `argument for the current row / sum of` ratio_expression `argument for the window or partition`

Il seguente insieme di dati dimostra l'uso di questa formula:

```
Row# Value Calculation RATIO_TO_REPORT
1 2500 (2500)/(13900) 0.1798
2 2600 (2600)/(13900) 0.1870
3 2800 (2800)/(13900) 0.2014
4 2900 (2900)/(13900) 0.2086
5 3100 (3100)/(13900) 0.2230
```

L'intervallo del valore di restituzione è compreso tra 0 e 1, con questi valori compresi. Se ratio_expression è NULL, allora il valore restituito è NULL. Se un valore in partition_expression è unico, la funzione restituirà 1 per quel valore.

Sintassi

```
RATIO_TO_REPORT ( ratio_expression )
OVER ( [ PARTITION BY partition_expression ] )
```

Argomenti

ratio_expression

Un'espressione, come ad esempio un nome di colonna, che fornisce il valore per cui determinare il rapporto. L'espressione deve avere o un tipo di dati numerici o essere implicitamente convertibile in uno.

Non è possibile utilizzare altre funzioni analitiche in ratio_expression.

OVER

Una clausola che specifica il partizionamento della finestra. La clausola OVER non può contenere una specifica del frame della finestra o dell'ordinamento della finestra.

PARTITION BY partition_expression

Facoltativo. Un'espressione che imposta l'intervallo di registrazioni per ciascun gruppo nella clausola OVER.

Tipo restituito

FLOAT8

Esempi

Gli esempi seguenti usano la tabella WINSALES. Per informazioni su come creare la tabella WINSALES, consulta Tabella di esempio per gli esempi della funzione finestra.

L'esempio seguente calcola il ratio-to-report valore di ogni riga della quantità di un venditore rispetto al totale di tutte le quantità del venditore.

```
select sellerid, qty, ratio_to_report(qty)
over()
from winsales
order by sellerid;

sellerid  qty    ratio_to_report
-------------------------------------
1         30     0.13953488372093023
1         10     0.046511627906976744
1         10     0.046511627906976744
2         20     0.09302325581395349
2         20     0.09302325581395349
3         30     0.13953488372093023
3         20     0.09302325581395349
3         15     0.06976744186046512
3         10     0.046511627906976744
4         10     0.046511627906976744
4         40     0.18604651162790697
```

L'esempio seguente calcola i rapporti delle quantità di vendita per ciascun venditore per partizione:

```
select sellerid, qty, ratio_to_report(qty)
over(partition by sellerid)
from winsales;
```

```
sellerid   qty    ratio_to_report
-----------------------------------------
2          20     0.5
2          20     0.5
4          40     0.8
4          10     0.2
1          10     0.2
1          30     0.6
1          10     0.2
3          10     0.13333333333333333
3          15     0.2
3          20     0.26666666666666666
3          30     0.4
```

Funzione finestra ROW_NUMBER

Assegna un numero ordinale di una riga corrente in un gruppo di righe, contando da 1, in base all'espressione ORDER BY nella clausola OVER. Se è presente la clausola PARTITION BY facoltativa, i numeri ordinali vengono ripristinati per ciascun gruppo di righe. Le righe con valori uguali per le espressioni ORDER BY ricevono i numeri di riga diversi in modo non deterministico.

Sintassi

```
ROW_NUMBER() OVER(
  [ PARTITION BY expr_list ]
  [ ORDER BY order_list ]
)
```

Argomenti

( )

La funzione non accetta argomenti, ma le parentesi vuote sono obbligatorie.

OVER

La clausola finestra per la funzione ROW_NUMBER.

PARTITION BY expr_list

Facoltativo. Una o più espressioni di colonna che dividono i risultati in serie di righe.

ORDER BY order_list

Facoltativo. Una o più espressioni di colonna che definiscono l'ordine delle righe all'interno di un set. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella.

Se ORDER BY non produce un ordinamento univoco o viene omesso, l'ordine delle righe non è deterministico. Per ulteriori informazioni, consultare Ordinamento univoco dei dati per le funzioni finestra.

Tipo restituito

BIGINT

Esempi

Gli esempi seguenti usano la tabella `WINSALES`. Per una descrizione della tabella `WINSALES`, consulta Tabella di esempio per gli esempi della funzione finestra.

L'esempio seguente esegue ordina la tabella in base a QTY (in ordine crescente), quindi assegna un numero di riga a ogni riga. I risultati vengono ordinati dopo aver applicato i risultati della funzione finestra.

```
SELECT salesid, sellerid, qty,
ROW_NUMBER() OVER(
   ORDER BY qty ASC) AS row
FROM winsales
ORDER BY 4,1;

salesid   sellerid   qty   row
---------+----------+-----+-----
   30001 |        3 |  10 |   1
   10001 |        1 |  10 |   2
   10006 |        1 |  10 |   3
   40005 |        4 |  10 |   4
   30003 |        3 |  15 |   5
   20001 |        2 |  20 |   6
   20002 |        2 |  20 |   7
   30004 |        3 |  20 |   8
   10005 |        1 |  30 |   9
   30007 |        3 |  30 |  10
   40001 |        4 |  40 |  11
```

L'esempio seguente esegue la partizione della tabella in SELLERID e ordina ciascuna partizione in base a QTY (in ordine crescente), quindi assegna un numero di riga a ogni riga. I risultati vengono ordinati dopo aver applicato i risultati della funzione finestra.

```
SELECT salesid, sellerid, qty,
ROW_NUMBER() OVER(
  PARTITION BY sellerid
  ORDER BY qty ASC) AS row_by_seller
FROM winsales
ORDER BY 2,4;

 salesid | sellerid | qty | row_by_seller
---------+----------+-----+-----
   10001 |        1 |  10 |   1
   10006 |        1 |  10 |   2
   10005 |        1 |  30 |   3
   20001 |        2 |  20 |   1
   20002 |        2 |  20 |   2
   30001 |        3 |  10 |   1
   30003 |        3 |  15 |   2
   30004 |        3 |  20 |   3
   30007 |        3 |  30 |   4
   40005 |        4 |  10 |   1
   40001 |        4 |  40 |   2
```

L'esempio seguente mostra i risultati quando non si utilizzano le clausole opzionali.

```
SELECT salesid, sellerid, qty, ROW_NUMBER() OVER() AS row
FROM winsales
ORDER BY 4,1;

salesid   sellerid   qty   row
---------+----------+-----+-----
   30001 |        3 |  10 |   1
   10001 |        1 |  10 |   2
   10005 |        1 |  30 |   3
   40001 |        4 |  40 |   4
   10006 |        1 |  10 |   5
   20001 |        2 |  20 |   6
   40005 |        4 |  10 |   7
   20002 |        2 |  20 |   8
   30003 |        3 |  15 |   9
   30004 |        3 |  20 |  10
```

```
  30007 |        3 |  30 |  11
```

Funzioni finestra STDDEV_SAMP e STDDEV_POP

Le funzioni finestra STDDEV_SAMP e STDDEV_POP restituiscono la deviazione standard del campione e della popolazione di un insieme di valori numerici (integer, numero decimale, numero in virgola mobile). consultare anche Funzioni STDDEV_SAMP e STDDEV_POP.

STDDEV_SAMP e STDDEV sono sinonimi della stessa funzione.

Sintassi

```
STDDEV_SAMP | STDDEV | STDDEV_POP
( [ ALL ] expression ) OVER
(
[ PARTITION BY expr_list ]
[ ORDER BY order_list
                        frame_clause ]
)
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

ALL

Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione. ALL è il valore predefinito. DISTINCT non è supportato.

OVER

Specifica le clausole finestra per le funzioni di aggregazione. La clausola OVER distingue le funzioni di aggregazione delle finestre dalle normali funzioni di aggregazione dell'insieme.

PARTITION BY expr_list

Definisce la finestra per la funzione in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe all'interno del risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipi di dati

I tipi di argomenti supportati dalle funzioni STDDEV sono SMALLINT, INTEGER, BIGINT, NUMERIC, DECIMAL, REAL e DOUBLE PRECISION.

Indipendentemente dal tipo di dati dell'espressione, il tipo di restituzione di una funzione STDDEV è un numero a precisione doppia.

Esempi

L'esempio seguente mostra come utilizzare le funzioni STDDEV_POP e VAR_POP come funzioni della finestra. La query calcola la variazione della popolazione e la deviazione standard della popolazione per i valori PRICEPAID nella tabella SALES.

```
select salesid, dateid, pricepaid,
round(stddev_pop(pricepaid) over
(order by dateid, salesid rows unbounded preceding)) as stddevpop,
round(var_pop(pricepaid) over
(order by dateid, salesid rows unbounded preceding)) as varpop
from sales
order by 2,1;

salesid | dateid | pricepaid | stddevpop | varpop
--------+--------+-----------+-----------+---------
  33095 |   1827 |    234.00 |         0 |       0
  65082 |   1827 |    472.00 |       119 |   14161
  88268 |   1827 |    836.00 |       248 |   61283
  97197 |   1827 |    708.00 |       230 |   53019
 110328 |   1827 |    347.00 |       223 |   49845
 110917 |   1827 |    337.00 |       215 |   46159
 150314 |   1827 |    688.00 |       211 |   44414
 157751 |   1827 |   1730.00 |       447 |  199679
 165890 |   1827 |   4192.00 |      1185 | 1403323
```

```
...
```

Le funzioni di deviazione standard e di varianza del campione possono essere utilizzate allo stesso modo.

Funzione finestra SUM

La funzione finestra SUM restituisce la somma dei valori dell'espressione o della colonna di input. La funzione SUM funziona con i valori numerici e ignora i valori NULL.

Sintassi

```
SUM ( [ ALL ] expression ) OVER
(
[ PARTITION BY expr_list ]
[ ORDER BY order_list
                        frame_clause ]
)
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

ALL

Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione. ALL è il valore predefinito. DISTINCT non è supportato.

OVER

Specifica le clausole finestra per le funzioni di aggregazione. La clausola OVER distingue le funzioni di aggregazione delle finestre dalle normali funzioni di aggregazione dell'insieme.

PARTITION BY expr_list

Definisce la finestra per la funzione SUM in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe all'interno del risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipi di dati

I tipi di argomenti supportati dalla funzione SUM sono SMALLINT, INTEGER, BIGINT, NUMERIC, DECIMAL, REAL e DOUBLE PRECISION.

I tipi di restituzione supportati dalla funzione SUM sono:

- BIGINT per gli argomenti SMALLINT oppure INTEGER
- NUMERIC per gli argomenti BIGINT
- DOUBLE PRECISION per argomenti del numero in virgola mobile

Esempi

Nel seguente esempio viene creata una somma cumulativa (rolling) delle quantità di vendita ordinate per data e ID vendite:

```
select salesid, dateid, sellerid, qty,
sum(qty) over (order by dateid, salesid rows unbounded preceding) as sum
from winsales
order by 2,1;

salesid |   dateid   | sellerid | qty | sum
---------+------------+----------+-----+-----
30001 | 2003-08-02 |        3 |  10 |  10
10001 | 2003-12-24 |        1 |  10 |  20
10005 | 2003-12-24 |        1 |  30 |  50
40001 | 2004-01-09 |        4 |  40 |  90
10006 | 2004-01-18 |        1 |  10 | 100
20001 | 2004-02-12 |        2 |  20 | 120
40005 | 2004-02-12 |        4 |  10 | 130
20002 | 2004-02-16 |        2 |  20 | 150
30003 | 2004-04-18 |        3 |  15 | 165
30004 | 2004-04-18 |        3 |  20 | 185
```

```
30007 | 2004-09-07 |        3 |  30 | 215
(11 rows)
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Nel seguente esempio viene creata una somma cumulativa (rolling) delle quantità di vendita per data, i risultati sono partizionati per ID venditore e all'interno della partizione i risultati sono ordinati per data e ID vendite:

```
select salesid, dateid, sellerid, qty,
sum(qty) over (partition by sellerid
order by dateid, salesid rows unbounded preceding) as sum
from winsales
order by 2,1;

salesid |   dateid   | sellerid | qty | sum
---------+------------+----------+-----+-----
30001 | 2003-08-02 |        3 |  10 |  10
10001 | 2003-12-24 |        1 |  10 |  10
10005 | 2003-12-24 |        1 |  30 |  40
40001 | 2004-01-09 |        4 |  40 |  40
10006 | 2004-01-18 |        1 |  10 |  50
20001 | 2004-02-12 |        2 |  20 |  20
40005 | 2004-02-12 |        4 |  10 |  50
20002 | 2004-02-16 |        2 |  20 |  40
30003 | 2004-04-18 |        3 |  15 |  25
30004 | 2004-04-18 |        3 |  20 |  45
30007 | 2004-09-07 |        3 |  30 |  75
(11 rows)
```

Nel seguente esempio sono numerate tutte le righe nel set di risultati, ordinate dalle colonne SELLERID e SALESID:

```
select salesid, sellerid, qty,
sum(1) over (order by sellerid, salesid rows unbounded preceding) as rownum
from winsales
order by 2,1;

salesid | sellerid |  qty | rownum
--------+----------+------+--------
10001 |        1 |   10 |     1
```

```
10005 |        1 |   30 |     2
10006 |        1 |   10 |     3
20001 |        2 |   20 |     4
20002 |        2 |   20 |     5
30001 |        3 |   10 |     6
30003 |        3 |   15 |     7
30004 |        3 |   20 |     8
30007 |        3 |   30 |     9
40001 |        4 |   40 |    10
40005 |        4 |   10 |    11
(11 rows)
```

Per una descrizione della tabella WINSALES, consultare Tabella di esempio per gli esempi della funzione finestra.

Nel seguente esempio vengono numerate tutte le righe nel set di risultati e i risultati sono partizionati per SELLERID e ordinati per SELLERID e SALESID e all'interno della partizione:

```
select salesid, sellerid, qty,
sum(1) over (partition by sellerid
order by sellerid, salesid rows unbounded preceding) as rownum
from winsales
order by 2,1;

salesid | sellerid | qty | rownum
---------+----------+-----+--------
10001 |        1 |  10 |      1
10005 |        1 |  30 |      2
10006 |        1 |  10 |      3
20001 |        2 |  20 |      1
20002 |        2 |  20 |      2
30001 |        3 |  10 |      1
30003 |        3 |  15 |      2
30004 |        3 |  20 |      3
30007 |        3 |  30 |      4
40001 |        4 |  40 |      1
40005 |        4 |  10 |      2
(11 rows)
```

Funzioni finestra VAR_SAMP e VAR_POP

Le funzioni finestra VAR_SAMP e VAR_POP restituiscono la varianza del campione e della popolazione di un insieme di valori numerici (integer, numero decimale, numero in virgola mobile). consultare anche Funzioni VAR_SAMP e VAR_POP.

VAR_SAMP e VARIANCE sono sinonimi della stessa funzione.

Sintassi

```
VAR_SAMP | VARIANCE | VAR_POP
( [ ALL ] expression ) OVER
(
[ PARTITION BY expr_list ]
[ ORDER BY order_list
                        frame_clause ]
)
```

Argomenti

expression

L'espressione o colonna di destinazione su cui viene eseguita la funzione.

ALL

Con l'argomento ALL, la funzione mantiene tutti i valori duplicati dall'espressione. ALL è il valore predefinito. DISTINCT non è supportato.

OVER

Specifica le clausole finestra per le funzioni di aggregazione. La clausola OVER distingue le funzioni di aggregazione delle finestre dalle normali funzioni di aggregazione dell'insieme.

PARTITION BY expr_list

Definisce la finestra per la funzione in termini di una o più espressioni.

ORDER BY order_list

Ordina le righe all'interno di ogni partizione. Se non viene specificato nessun PARTITION BY, ORDER BY utilizza l'intera tabella.

frame_clause

Se una clausola ORDER BY viene utilizzata per una funzione di aggregazione, è necessaria una clausola del frame esplicita. La clausola frame raffina l'insieme di righe in una finestra della funzione, includendo o escludendo insieme di righe all'interno del risultato ordinato. La clausola frame è composta dalla parola chiave ROWS e dagli specificatori associati. Per informazioni, consultare Riepilogo della sintassi della funzione finestra.

Tipi di dati

I tipi di argomenti supportati dalle funzioni VARIANCE sono SMALLINT, INTEGER, BIGINT, NUMERIC, DECIMAL, REAL e DOUBLE PRECISION.

Indipendentemente dal tipo di dati dell'espressione, il tipo di restituzione di una funzione VARIANCE è un numero a precisione doppia.

Funzioni di amministrazione del sistema

Argomenti

Amazon Redshift supporta diverse funzioni di amministrazione del sistema.

CHANGE_QUERY_PRIORITY

CHANGE_QUERY_PRIORITY permette agli utenti con privilegi avanzati di modificare la priorità di una query in esecuzione o in attesa nella gestione del carico di lavoro (workload management, WLM).

Questa funzione permette agli utenti con privilegi avanzati di modificare immediatamente la priorità di qualunque query nel sistema. Solo una query, un utente o una sessione possono essere eseguiti con la priorità `CRITICAL`.

Sintassi

```
CHANGE_QUERY_PRIORITY(query_id, priority)
```

Argomenti

query_id

L'identificatore della query di cui è cambiata la priorità. Richiede un valore `INTEGER`.

priority

La nuova priorità da assegnare alla query. Questo argomento deve essere una stringa con il valore `CRITICAL`, `HIGHEST`, `HIGH`, `NORMAL`, `LOW` o `LOWEST`.

Tipo restituito

Nessuno

Esempi

Per mostrare la colonna `query_priority` nella tabella di sistema STV_WLM_QUERY_STATE, utilizza l'esempio seguente.

```
SELECT query, service_class, query_priority, state
FROM stv_wlm_query_state WHERE service_class = 101;

+-------+---------------+----------------+---------+
| query | service_class | query_priority |  state  |
+-------+---------------+----------------+---------+
|  1076 |           101 | Lowest         | Running |
|  1075 |           101 | Lowest         | Running |
+-------+---------------+----------------+---------+
```

Per mostrare i risultati di un utente con privilegi avanzati che esegue la funzione `change_query_priority` per modificare la priorità in `CRITICAL`, utilizza l'esempio seguente.

```
SELECT CHANGE_QUERY_PRIORITY(1076, 'Critical');
```

```
+-------------------------------------------------------------------------------+
|                            change_query_priority                              |
+-------------------------------------------------------------------------------+
| Succeeded to change query priority. Priority changed from Lowest to Critical. |
+-------------------------------------------------------------------------------+
```

CHANGE_SESSION_PRIORITY

CHANGE_SESSION_PRIORITY permette agli utenti con privilegi avanzati di modificare immediatamente la priorità di qualunque sessione nel sistema. Solo una sessione, un utente o una query possono essere eseguiti con la priorità `CRITICAL`.

Sintassi

```
CHANGE_SESSION_PRIORITY(pid, priority)
```

Argomenti

pid

L'identificatore processo della sessione di cui è cambiata la priorità. Il valore `-1` si riferisce alla sessione attuale. Richiede un valore `INTEGER`.

priority

La nuova priorità da assegnare alla sessione. Questo argomento deve essere una stringa con il valore `CRITICAL`, `HIGHEST`, `HIGH`, `NORMAL`, `LOW` o `LOWEST`.

Tipo restituito

Nessuno

Esempi

Per restituire l'identificatore del processo del server che gestisce la sessione corrente, utilizza l'esempio seguente.

```
SELECT pg_backend_pid();

+----------------+
| pg_backend_pid |
```

```
+----------------+
|          30311 |
+----------------+
```

In questo esempio, la priorità viene modificata in `LOWEST` per la sessione attuale.

```
SELECT CHANGE_SESSION_PRIORITY(30311, 'Lowest');

+-------------------------------------------------------------------------------
+
|                              change_session_priority
 |
+-------------------------------------------------------------------------------
+
| Succeeded to change session priority. Changed session (pid:30311) priority to lowest.
 |
+-------------------------------------------------------------------------------
+
```

In questo esempio, la priorità viene modificata in `HIGH` per la sessione attuale.

```
SELECT CHANGE_SESSION_PRIORITY(-1, 'High');

+-------------------------------------------------------------------------------
+
|                                  change_session_priority
           |
+-------------------------------------------------------------------------------
+
| Succeeded to change session priority. Changed session (pid:30311) priority from
 lowest to high. |
+-------------------------------------------------------------------------------
+
```

Per creare una stored procedure per modificare la priorità di una sessione, utilizza l'esempio seguente. L'autorizzaione per eseguire questa procedura archiviata viene concessa all'utente di database `test_user`.

```
CREATE OR REPLACE PROCEDURE sp_priority_low(pid IN int, result OUT varchar)
AS $$
BEGIN
  SELECT CHANGE_SESSION_PRIORITY(pid, 'low') into result;
```

```
END;
$$ LANGUAGE plpgsql
SECURITY DEFINER;
GRANT EXECUTE ON PROCEDURE sp_priority_low(int) TO test_user;
```

Quindi, l'utente di database denominato `test_user` richiama la procedura.

```
CALL sp_priority_low(pg_backend_pid());

+-----------------------------------------------------+
|                        result                       |
+-----------------------------------------------------+
| Success. Change session (pid:13155) priority to low. |
+-----------------------------------------------------+
```

CHANGE_USER_PRIORITY

CHANGE_SESSION_PRIORITY permette agli utente con privilegi avanzati di modificare la priorità tutte le query inviata da un utente, in esecuzione o in attesa in gestione del carico di lavoro (WLM). Solo un utente, una sessione o una query possono essere eseguiti con la priorità `CRITICAL`.

Sintassi

```
CHANGE_USER_PRIORITY(user_name, priority)
```

Argomenti

user_name

Il nome dell'utente di database di cui è cambiata la priorità di query.

priority

La nuova priorità da assegnare a tutte le query inviate da `user_name`. Questo argomento deve consistere in una stringa con valore `CRITICAL`, `HIGHEST`, `HIGH`, `NORMAL`, `LOW`, `LOWEST` o `RESET`. Solo gli utenti con privilegi avanzati possono modificare la priorità in `CRITICAL`. Modificare la priorità in `RESET` rimuove l'impostazione di priorità per `user_name`.

Tipo restituito

Nessuno

Esempi

Per modificare la priorità dell'utente `analysis_user` in `LOWEST`, utilizza l'esempio seguente.

```
SELECT CHANGE_USER_PRIORITY('analysis_user', 'lowest');

+-------------------------------------------------------------------------------------+
|                               change_user_priority                                  |
+-------------------------------------------------------------------------------------+
| Succeeded to change user priority. Changed user (analysis_user) priority to lowest. |
+-------------------------------------------------------------------------------------+
```

Per modificare la priorità in `LOW`, utilizza l'esempio seguente.

```
SELECT CHANGE_USER_PRIORITY('analysis_user', 'low');

+-------------------------------------------------------------------------------------------------------------
+
|                                    change_user_priority
        |
+-------------------------------------------------------------------------------------------------------------
+
| Succeeded to change user priority. Changed user (analysis_user) priority from Lowest
 to low. |
+-------------------------------------------------------------------------------------------------------------
+
```

Per reimpostare la priorità, utilizza l'esempio seguente.

```
SELECT CHANGE_USER_PRIORITY('analysis_user', 'reset');

+-------------------------------------------------------+
|                change_user_priority                   |
+-------------------------------------------------------+
| Succeeded to reset priority for user (analysis_user). |
+-------------------------------------------------------+
```

CURRENT_SETTING

CURRENT_SETTING restituisce il valore corrente del parametro di configurazione specificato.

Questa funzione è equivalente al comando MOSTRA.

Sintassi

```
current_setting('parameter')
```

L'istruzione seguente restituisce il valore corrente di una variabile di contesto di sessione.

```
current_setting('variable_name')
current_setting('variable_name'[, error_if_undefined])
```

Argomenti

parameter

Valore del parametro da visualizzare. Per un elenco dei parametri di configurazione, consultare Informazioni di riferimento sulla configurazione

variable_name

Il nome della variabile da visualizzare. Deve essere una costante di stringa per le variabili di contesto di sessione.

error_if_undefined

(Facoltativo) Un valore booleano che specifica il comportamento se il nome della variabile non esiste. Quando error_if_undefined è impostato su TRUE, che è l'impostazione predefinita, Amazon Redshift genera un errore. Quando error_if_undefined è impostato su FALSE, Amazon Redshift restituisce NULL. Amazon Redshift supporta il parametro error_if_undefined solo per le variabili di contesto di sessione. Ciò non può essere utilizzato quando l'input è un parametro di configurazione.

Tipo restituito

Restituisce CHAR o una stringa VARCHAR.

Esempi

Per restituire l'impostazione corrente per il parametro query_group, utilizza l'esempio seguente.

```
SELECT CURRENT_SETTING('query_group');

+-----------------+
```

```
| current_setting |
+-----------------+
| unset           |
+-----------------+
```

Per restituire l'impostazione corrente per la variabile `app_context.user_id`, utilizza l'esempio seguente.

```
SELECT CURRENT_SETTING('app_context.user_id', FALSE);
```

PG_CANCEL_BACKEND

Annulla una query. PG_CANCEL_BACKEND è funzionalmente equivalente al comando CANCEL. È possibile annullare solo le query attualmente eseguite dall'utente. Gli utenti con privilegi avanzati possono cancellare qualsiasi query.

Sintassi

```
pg_cancel_backend( pid )
```

Argomenti

pid

L'ID del processo (PID) della query da annullare. Non è possibile annullare una query specificando un ID query; è necessario specificare l'ID processo della query. Richiede un valore `INTEGER`.

Tipo restituito

Nessuno

Note per l'utilizzo

Se le query in più sessioni contengono blocchi sulla stessa tabella, è possibile utilizzare la funzione PG_TERMINATE_BACKEND per terminare una delle sessioni, forzando qualsiasi transazione attualmente in esecuzione nella sessione terminata per rilasciare tutti i blocchi ed eseguire il rollback della transazione. Eseguire una query sulla tabella del catalogo PG__LOCKS per visualizzare i blocchi attuali. Se non è possibile annullare una query perché si trova nel blocco di transazioni

(BEGIN ... END), è possibile interrompere la sessione in cui viene eseguita la query utilizzando la funzione PG_TERMINATE_BACKEND.

Esempi

Per annullare una query attualmente in esecuzione, recupera prima l'ID processo della query che vuoi annullare. Per determinare gli ID processo di tutte le query attualmente in esecuzione, esegui il seguente comando.

```
SELECT pid, TRIM(starttime) AS start,
duration, TRIM(user_name) AS user,
SUBSTRING(query,1,40) AS querytxt
FROM stv_recents
WHERE status = 'Running';

+-----+----------------------+----------+--------+----------------------------+
| pid |       starttime      | duration |  user  |          querytxt          |
+-----+----------------------+----------+--------+----------------------------+
| 802 | 2013-10-14 09:19:03.55 |      132 | dwuser | select venuename from venue |
| 834 | 2013-10-14 08:33:49.47 |  1250414 | dwuser | select * from listing;      |
| 964 | 2013-10-14 08:30:43.29 |   326179 | dwuser | select sellerid from sales  |
+-----+----------------------+----------+--------+----------------------------+
```

Per annullare la query con l'ID di processo 802, utilizza l'esempio seguente.

```
SELECT PG_CANCEL_BACKEND(802);
```

PG_TERMINATE_BACKEND

Interrompe una sessione. È possibile terminare una sessione di proprietà dell'utente. Un utente con privilegi avanzati può terminare qualsiasi sessione.

Sintassi

```
pg_terminate_backend( pid )
```

Argomenti

pid

L'ID di processo della sessione da interrompere. Richiede un valore `INTEGER`.

Tipo restituito

Nessuno

Note per l'utilizzo

Se si è prossimi a raggiungere il limite per le connessioni simultanee, usare PG_TERMINATE_BACKEND per interrompere le sessioni inattive e liberare le connessioni. Per ulteriori informazioni, consultare Limiti di Amazon Redshift.

Se le query in più sessioni contengono blocchi sulla stessa tabella, è possibile utilizzare la funzione PG_TERMINATE_BACKEND per terminare una delle sessioni, forzando qualsiasi transazione attualmente in esecuzione nella sessione terminata per rilasciare tutti i blocchi ed eseguire il rollback della transazione. Esegui una query sulla tabella del catalogo PG__LOCKS per visualizzare i blocchi attuali.

Se una query non si trova in un blocco di transazione (BEGIN ... END), è possibile annullare la query utilizzando il comando CANCEL oppure la funzione PG_CANCEL_BACKEND.

Esempi

Per eseguire query sulla tabella SVV_TRANSACTIONS per visualizzare tutti i blocchi attivi per le transazioni correnti, utilizza l'esempio seguente.

```
SELECT * FROM svv_transactions;

+----------+--------+-------+------+--------------------+-----------------
+----------------------+----------+---------+
| txn_owner | txn_db |  xid  | pid  |      txn_start      |    lock_mode    |
 lockable_object_type | relation | granted |
+----------+--------+-------+------+--------------------+-----------------
+----------------------+----------+---------+
| rsuser    | dev    | 96178 | 8585 | 2017-04-12 20:13:07 | AccessShareLock | relation
           |    51940 | true    |
| rsuser    | dev    | 96178 | 8585 | 2017-04-12 20:13:07 | AccessShareLock | relation
           |    52000 | true    |
| rsuser    | dev    | 96178 | 8585 | 2017-04-12 20:13:07 | AccessShareLock | relation
           |   108623 | true    |
| rsuser    | dev    | 96178 | 8585 | 2017-04-12 20:13:07 | ExclusiveLock   |
 transactionid        |          | true    |
+----------+--------+-------+------+--------------------+-----------------
+----------------------+----------+---------+
```

PER terminare la sessione con i blocchi, utilizza l'esempio seguente.

```
SELECT PG_TERMINATE_BACKEND(8585);
```

REBOOT_CLUSTER

Riavvia il cluster Amazon Redshift senza chiudere le connessioni al cluster. Per eseguire questo comando, è necessario essere un utente con privilegi avanzati del database.

Al termine del riavvio soft, il cluster Amazon Redshift restituisce un errore all'applicazione utente e richiede all'applicazione utente di inviare nuovamente le transazioni o le query interrotte dal riavvio.

Sintassi

```
SELECT REBOOT_CLUSTER();
```

SET_CONFIG

Imposta un parametro di configurazione su una nuova impostazione.

Questa funzione è equivalente al comando SET in SQL.

Sintassi

```
SET_CONFIG('parameter', 'new_value' , is_local)
```

La seguente istruzione fornisce una nuova impostazione a una variabile di contesto di sessione.

```
set_config('variable_name', 'new_value' , is_local)
```

Argomenti

parameter

Parametro da impostare.

variable_name

Il nome della variabile da impostare.

new_value

Nuovo valore del parametro.

is_local

Se vero, il valore del parametro si applica solo alla transazione corrente. I valori validi sono `true` o `1` e `false` o `0`.

Tipo restituito

Restituisce CHAR o una stringa VARCHAR.

Esempi

Per impostare il valore del parametro `query_group` in `test` solo per la transazione corrente, utilizza l'esempio seguente.

```
SELECT SET_CONFIG('query_group', 'test', true);

+------------+
| set_config |
+------------+
| test       |
+------------+
```

Per impostare le variabili di contesto di sessione, utilizza l'esempio seguente.

```
SELECT SET_CONFIG('app.username', 'cuddy', FALSE);
```

Funzioni di informazioni di sistema

Amazon Redshift supporta numerose funzioni informative del sistema.

Argomenti

- CURRENT_AWS_ACCOUNT
- CURRENT_DATABASE
- CURRENT_NAMESPACE
- CURRENT_SCHEMA

- CURRENT_SCHEMAS
- CURRENT_USER
- CURRENT_USER_ID
- DEFAULT_IAM_ROLE
- HAS_ASSUMEROLE_PRIVILEGE
- HAS_DATABASE_PRIVILEGE
- PRIVILEGIO DELLO SCHEMA HAS
- HAS_TABLE_PRIVILEGE
- LAST_USER_QUERY_ID
- PG_BACKEND_PID
- PG_GET_COLS
- PG_GET_GRANTEE_BY_IAM_ROLE
- PG_GET_IAM_ROLE_BY_USER
- PG_GET_LATE_BINDING_VIEW_COLS
- PG_GET_SESSION_ROLES
- PG_LAST_COPY_COUNT
- PG_LAST_COPY_ID
- PG_LAST_UNLOAD_ID
- PG_LAST_QUERY_ID
- PG_LAST_UNLOAD_COUNT
- SLICE_NUM Function
- UTENTE
- ROLE_IS_MEMBER_OF
- USER_IS_MEMBER_OF
- VERSION

CURRENT_AWS_ACCOUNT

Restituisce l' AWS account associato al cluster Amazon Redshift che ha inviato una query.

Sintassi

```
current_aws_account
```

Tipo restituito

Restituisce un integer.

Esempio

La seguente query restituisce il nome del database corrente:

```
select user, current_aws_account;
current_user | current_account
-------------+---------------
dwuser       | 987654321

(1 row)
```

CURRENT_DATABASE

Restituisce il nome del database in cui si è attualmente connessi.

Sintassi

```
current_database()
```

Tipo restituito

Restituisce una stringa CHAR o VARCHAR.

Esempio

La seguente query restituisce il nome del database corrente:

```
select current_database();

current_database
-----------------
tickit
```

```
(1 row)
```

CURRENT_NAMESPACE

Restituisce lo spazio dei nomi del cluster Amazon Redshift corrente. Lo spazio dei nomi del cluster Amazon Redshift è l'ID univoco del cluster Amazon Redshift.

Sintassi

```
current_namespace
```

Tipo restituito

Restituisce una stringa CHAR o VARCHAR.

Esempio

La seguente query restituisce il nome dello spazio dei nomi corrente.

```
select user, current_namespace;
current_user | current_namespace
-------------+--------------------------------------
dwuser       | 86b5169f-01dc-4a6f-9fbb-e2e24359e9a8

(1 row)
```

CURRENT_SCHEMA

Restituisce il nome dello schema nella parte anteriore del percorso di ricerca. Questo schema verrà utilizzato per tutte le tabelle o altri oggetti denominati che vengono creati senza specificare uno schema di destinazione.

Sintassi

> **Note**
>
> Questa è una funzione del nodo principale. Questa funzione restituisce un errore se fa riferimento a una tabella creata dall'utente, a una tabella di sistema STL o STV o a una vista di sistema SVV o SVL.

```
current_schema()
```

Tipo restituito

CURRENT_SCHEMA restituisce una stringa CHAR o VARCHAR.

Esempi

La seguente query restituisce lo schema attuale:

```
select current_schema();

current_schema
---------------
public
(1 row)
```

CURRENT_SCHEMAS

Restituisce un array dei nomi di tutti gli schemi nel percorso di ricerca corrente. Il percorso di ricerca corrente viene definito nel parametro search_path.

Sintassi

> **Note**
>
> Questa è una funzione del nodo principale. Questa funzione restituisce un errore se fa riferimento a una tabella creata dall'utente, a una tabella di sistema STL o STV o a una vista di sistema SVV o SVL.

```
current_schemas(include_implicit)
```

Argomento

include_implicit

Se vero, specifica che il percorso di ricerca deve includere tutti gli schemi di sistema inclusi implicitamente. I valori validi sono `true` e `false`. In genere, se `true`, questo parametro restituisce lo schema `pg_catalog` oltre allo schema attuale.

Tipo restituito

Restituisce una stringa CHAR o VARCHAR.

Esempi

L'esempio seguente restituisce i nomi degli schemi nel percorso di ricerca corrente, esclusi gli schemi di sistema inclusi implicitamente:

```
select current_schemas(false);

current_schemas
----------------
{public}
(1 row)
```

L'esempio seguente restituisce i nomi degli schemi nel percorso di ricerca corrente, compresi gli schemi di sistema inclusi implicitamente:

```
select current_schemas(true);

current_schemas
--------------------
{pg_catalog,public}
(1 row)
```

CURRENT_USER

Restituisce il nome utente dell'utente corrente "effettivo" del database, come applicabile alle autorizzazioni di controllo. Di solito, questo nome utente sarà lo stesso dell'utente della sessione; tuttavia, questo può essere cambiato occasionalmente dagli utenti con privilegi avanzati.

> **Note**
>
> Non utilizzare parentesi finali durante la chiamata a CURRENT_USER.

Sintassi

```
current_user
```

Tipo restituito

CURRENT_USER restituisce un tipo di dati NAME e può essere convertito in una stringa CHAR o VARCHAR.

Note per l'utilizzo

Se una procedura archiviata è stata creata utilizzando l'opzione SECURITY DEFINER del comando CREATE_PROCEDURE, quando si richiama la funzione CURRENT_USER dall'interno della procedura archiviata, Amazon Redshift restituisce il nome utente del proprietario della procedura archiviata.

Esempio

La seguente query restituisce il nome dell'utente del database corrente:

```
select current_user;

current_user
--------------
dwuser
(1 row)
```

CURRENT_USER_ID

Restituisce l'identificativo univoco per l'utente di Amazon Redshift connesso alla sessione corrente.

Sintassi

```
CURRENT_USER_ID
```

Tipo restituito

La funzione CURRENT_USER_ID restituisce un integer.

Esempi

L'esempio seguente restituisce il nome utente e l'ID utente corrente per questa sessione:

```
select user, current_user_id;
```

```
 current_user | current_user_id
--------------+-----------------
   dwuser     |               1
(1 row)
```

DEFAULT_IAM_ROLE

Restituisce il ruolo IAM predefinito attualmente associato al cluster Amazon Redshift. La funzione non restituisce niente se non è associato alcun ruolo IAM predefinito.

Sintassi

```
select default_iam_role();
```

Tipo restituito

Restituisce una stringa VARCHAR.

Esempio

Il seguente esempio restituisce il ruolo IAM di default correntemente associato al cluster Amazon Redshift specificato.

```
select default_iam_role();
              default_iam_role
---------------------------------------------
 arn:aws:iam::123456789012:role/myRedshiftRole
(1 row)
```

HAS_ASSUMEROLE_PRIVILEGE

Restituisce Boolean `true` (`t`) se l'utente specificato ha l'utente IAM specificato con il privilegio per eseguire il comando specificato. La funzione restituisce `false` (`f`) se l'utente non ha l'utente IAM specificato con il privilegio per eseguire il comando specificato. Per ulteriori informazioni sui privilegi, consultare GRANT.

Sintassi

```
has_assumerole_privilege( [ user, ] iam_role_arn, cmd_type)
```

Argomenti

utente

Il nome dell'utente per controllare i privilegi dell'utente IAM. Il valore predefinito serve a controllare l'utente corrente. Questa funzione può essere utilizzata dagli utenti con privilegi avanzati e gli utenti. Tuttavia, gli utenti possono visualizzare solo i propri privilegi.

iam_role_arn

Il ruolo IAM a cui sono stati concessi i privilegi per il comando.

cmd_type

Il comando per il quale è stato concesso l'accesso. I valori validi sono i seguenti.

- COPY
- UNLOAD
- EXTERNAL FUNCTION
- CREATE MODEL

Tipo restituito

BOOLEAN

Esempio

La query seguente conferma che l'utente `reg_user1` dispone del privilegio per il ruolo `Redshift-S3-Read` per eseguire il comando COPY.

```
select has_assumerole_privilege('reg_user1', 'arn:aws:iam::123456789012:role/Redshift-S3-Read', 'copy');
```

```
has_assumerole_privilege
------------------------
true
(1 row)
```

HAS_DATABASE_PRIVILEGE

Restituisce `true` se l'utente ha il privilegio specificato per il database specificato. Per ulteriori informazioni sui privilegi, consultare GRANT.

Sintassi

> **Note**
>
> Questa è una funzione del nodo principale. Questa funzione restituisce un errore se fa riferimento a una tabella creata dall'utente, a una tabella di sistema STL o STV o a una vista di sistema SVV o SVL.

```
has_database_privilege( [ user, ] database, privilege)
```

Argomenti

utente

Il nome dell'utente per controllare i privilegi del database. Il valore predefinito serve a controllare l'utente corrente.

database

Il database associato al privilegio.

privilegio

Il privilegio da controllare. I valori validi sono i seguenti.

- CREATE
- TEMPORARY
- TEMP

Tipo restituito

Restituisce una stringa CHAR o VARCHAR.

Esempio

La seguente query conferma che l'utente GUEST ha il privilegio TEMP nel database TICKIT:

```
select has_database_privilege('guest', 'tickit', 'temp');

has_database_privilege
-----------------------
true
```

```
(1 row)
```

PRIVILEGIO DELLO SCHEMA HAS

Restituisce `true` se l'utente ha il privilegio specificato per lo schema specificato. Per ulteriori informazioni sui privilegi, consultare GRANT.

Sintassi

> **Note**
>
> Questa è una funzione del nodo principale. Questa funzione restituisce un errore se fa riferimento a una tabella creata dall'utente, a una tabella di sistema STL o STV o a una vista di sistema SVV o SVL.

```
has_schema_privilege( [ user, ] schema, privilege)
```

Argomenti

utente

Il nome dell'utente per controllare i privilegi dello schema. Il valore predefinito serve a controllare l'utente corrente.

schema

Lo schema associato al privilegio.

privilegio

Il privilegio da controllare. I valori validi sono i seguenti.

- CREATE
- USAGE

Tipo restituito

Restituisce una stringa CHAR o VARCHAR.

Esempio

La seguente query conferma che l'utente GUEST ha il privilegio CREATE nello schema PUBLIC:

```
select has_schema_privilege('guest', 'public', 'create');

has_schema_privilege
---------------------
true
(1 row)
```

HAS_TABLE_PRIVILEGE

Restituisce `true` se l'utente ha il privilegio specificato per la tabella specificata, altrimenti restituisce `false`.

Sintassi

> **Note**
>
> Questa è una funzione del nodo principale. Questa funzione restituisce un errore se fa riferimento a una tabella creata dall'utente, a una tabella di sistema STL o STV o a una vista di sistema SVV o SVL. Per ulteriori informazioni sui privilegi, consultare GRANT.

```
has_table_privilege( [ user, ] table, privilege)
```

Argomenti

utente

Il nome dell'utente per controllare i privilegi della tabella. Il valore predefinito serve a controllare l'utente corrente.

tabella

Tabella associata al privilegio.

privilegio

Privilegio da controllare. I valori validi sono i seguenti.

- SELECT
- INSERT
- UPDATE

- DELETE
- DROP
- REFERENCES

Tipo restituito

BOOLEAN

Esempi

La seguente query trova che l'utente GUEST non ha il privilegio SELECT nella tabella LISTING.

```
select has_table_privilege('guest', 'listing', 'select');

has_table_privilege
--------------------
false
```

La seguente query elenca i privilegi delle tabelle, inclusi select, insert, update e delete, utilizzando l'output delle tabelle del catalogo pg_tables e pg_user. Si tratta solo di un esempio, Potrebbe essere necessario specificare un nome di schema e i nomi delle tabelle del database. Per ulteriori informazioni, consulta Query sulle tabelle di catalogo.

```
SELECT
     tablename
     ,usename
     ,HAS_TABLE_PRIVILEGE(users.usename, tablename, 'select') AS sel
     ,HAS_TABLE_PRIVILEGE(users.usename, tablename, 'insert') AS ins
     ,HAS_TABLE_PRIVILEGE(users.usename, tablename, 'update') AS upd
     ,HAS_TABLE_PRIVILEGE(users.usename, tablename, 'delete') AS del
FROM
(SELECT * from pg_tables
WHERE schemaname = 'public' and tablename in ('event','listing')) as tables
,(SELECT * FROM pg_user) AS users;

tablename | usename   |  sel   |  ins  |  upd  | del
----------+-----------+--------+-------+-------+-------
event     |  john     |  true  | true  | true  | true
event     |  sally    |  false | false | false | false
event     |  elsa     |  false | false | false | false
listing   |  john     |  true  | true  | true  | true
```

```
listing   |  sally    |  false | false | false | false
listing   |  elsa     |  false | false | false | false
```

La query precedente contiene anche un cross join. Per ulteriori informazioni, consulta Esempi di JOIN. Per interrogare tabelle che non si trovano nello schema `public`, rimuovi la condizione `schemaname` dalla clausola WHERE e utilizza il seguente esempio prima della query.

```
SET SEARCH_PATH to 'schema_name';
```

LAST_USER_QUERY_ID

Restituisce l'ID della query di un utente completata più di recente nella sessione corrente. Se non sono state eseguite query nella sessione corrente, last_user_query_id restituisce -1. La funzione non restituisce l'ID della query per le query eseguite esclusivamente sul nodo principale. Per ulteriori informazioni, consulta Nodo principale: solo funzioni.

Sintassi

```
last_user_query_id()
```

Tipo restituito

Restituisce un integer.

Esempio

La seguente query restituisce l'ID dell'ultima query completata eseguita da un utente nella sessione corrente.

```
select last_user_query_id();
```

Di seguito sono riportati i risultati.

```
last_user_query_id
----------------------
    5437
(1 row)
```

La query seguente restituisce l'ID e il testo della query completata più di recente eseguita da un utente nella sessione corrente.

```
select query_id, query_text from sys_query_history where query_id =
 last_user_query_id();
```

Di seguito sono riportati i risultati.

```
 query_id, query_text
---------
+-----------------------------------------------------------------------------------------------------
 5556975 | select last_user_query_id() limit 100 --RequestID=<unique request ID>;
 TraceID=<unique trace ID>
```

PG_BACKEND_PID

Restituisce l'ID di processo (PID) del processo del server che gestisce la sessione corrente.

> **Note**
>
> L'PID non è univoco a livello globale. Può essere riutilizzato nel tempo.

Sintassi

```
pg_backend_pid()
```

Tipo restituito

Restituisce un integer.

Esempio

È possibile correlare PG_BACKEND_PID() con le tabelle di registro per recuperare le informazioni sulla sessione corrente. Ad esempio, la query seguente restituisce l'ID di query e una porzione del testo di query delle query completate nella sessione corrente.

```
select query, substring(text,1,40)
from stl_querytext
where pid =  PG_BACKEND_PID()
order by query desc;

 query |                substring
-------+----------------------------------------
```

```
 14831 | select query, substring(text,1,40) from
 14827 | select query, substring(path,0,80) as pa
 14826 | copy category from 's3://dw-tickit/manif
 14825 | Count rows in target table
 14824 | unload ('select * from category') to 's3
(5 rows)
```

È possibile correlare PG_BACKEND_PID con la colonna pid nelle seguenti tabelle di log (le eccezioni sono indicate tra parentesi):

- STL_CONNECTION_LOG
- STL_DDLTEXT
- STL_ERROR
- STL_QUERY
- STL_QUERYTEXT
- STL_SESSIONS (elaborazione)
- STL_TR_CONFLICT
- STL_UTILITYTEXT
- STV_ACTIVE_CURSORS
- STV_INFLIGHT
- STV_LOCKS (lock_owner_pid)
- STV_RECENTS (process_id)

PG_GET_COLS

Restituisce i metadati di colonna per una tabella o una definizione di visualizzazione.

Sintassi

```
pg_get_cols('name')
```

Argomenti

name

Il nome della vista o della tabella di Amazon Redshift. Per ulteriori informazioni, consultare Nomi e identificatori.

Tipo restituito

VARCHAR

Note per l'utilizzo

La funzione PG_GET_COLS restituisce una riga per ciascuna colonna nella tabella o nella definizione di visualizzazione. La riga contiene un elenco separato da virgole con il nome dello schema, il nome della relazione, il nome della colonna, il tipo di dati e il numero di colonna. La formattazione del risultato dell'istruzione SQL dipende dal client SQL utilizzato.

Esempi

Gli esempi seguenti restituiscono risultati per una vista denominata `SALES_VW` in schema `public` e una tabella denominata `sales` in schema `mytickit1` create dall'utente nel database `dev` connesso.

L'esempio seguente restituisce i metadati della colonna per una vista `SALES_VW` denominata.

```
select pg_get_cols('sales_vw');

pg_get_cols
---------------------------------------------------------
(public,sales_vw,salesid,integer,1)
(public,sales_vw,listid,integer,2)
(public,sales_vw,sellerid,integer,3)
(public,sales_vw,buyerid,integer,4)
(public,sales_vw,eventid,integer,5)
(public,sales_vw,dateid,smallint,6)
(public,sales_vw,qtysold,smallint,7)
(public,sales_vw,pricepaid,"numeric(8,2)",8)
(public,sales_vw,commission,"numeric(8,2)",9)
(public,sales_vw,saletime,"timestamp without time zone",10)
```

L'esempio seguente restituisce i metadati della colonna per la `SALES_VW` vista in formato tabella.

```
select * from pg_get_cols('sales_vw')
cols(view_schema name, view_name name, col_name name, col_type varchar, col_num int);

view_schema | view_name | col_name   | col_type                    | col_num
------------+-----------+------------+-----------------------------+--------
public      | sales_vw  | salesid    | integer                     |       1
public      | sales_vw  | listid     | integer                     |       2
```

```
public      | sales_vw  | sellerid   | integer                     |       3
public      | sales_vw  | buyerid    | integer                     |       4
public      | sales_vw  | eventid    | integer                     |       5
public      | sales_vw  | dateid     | smallint                    |       6
public      | sales_vw  | qtysold    | smallint                    |       7
public      | sales_vw  | pricepaid  | numeric(8,2)                |       8
public      | sales_vw  | commission | numeric(8,2)                |       9
public      | sales_vw  | saletime   | timestamp without time zone |      10
```

L'esempio seguente restituisce i metadati delle colonne per la SALES tabella nello schema `mytickit1` in formato tabella.

```
select * from pg_get_cols('"mytickit1"."sales"')
cols(view_schema name, view_name name, col_name name, col_type varchar, col_num int);

view_schema | view_name | col_name   | col_type                    | col_num
------------+-----------+------------+-----------------------------+--------
mytickit1   | sales     | salesid    | integer                     |       1
mytickit1   | sales     | listid     | integer                     |       2
mytickit1   | sales     | sellerid   | integer                     |       3
mytickit1   | sales     | buyerid    | integer                     |       4
mytickit1   | sales     | eventid    | integer                     |       5
mytickit1   | sales     | dateid     | smallint                    |       6
mytickit1   | sales     | qtysold    | smallint                    |       7
mytickit1   | sales     | pricepaid  | numeric(8,2)                |       8
mytickit1   | sales     | commission | numeric(8,2)                |       9
mytickit1   | sales     | saletime   | timestamp without time zone |      10
```

PG_GET_GRANTEE_BY_IAM_ROLE

Restituisce tutti gli utenti e i gruppi a cui è stato concesso un ruolo IAM specificato.

Sintassi

```
pg_get_grantee_by_iam_role('iam_role_arn')
```

Argomenti

iam_role_arn

Il ruolo IAM per il quale restituire gli utenti e i gruppi a cui è stato concesso questo ruolo.

Tipo restituito

VARCHAR

Note per l'utilizzo

La funzione PG_GET_GRANTEE_BY_IAM_ROLE restituisce una riga per ogni utente o gruppo. Ogni riga contiene il nome del concessionario, il tipo di assegnatario e il privilegio concesso. I valori possibili per il tipo beneficiario sono p per il pubblico, u per l'utente e g per il gruppo.

Per utilizzare questa funzione, è necessario essere un utente con privilegi avanzati.

Esempio

L'esempio seguente indica che il ruolo IAM `Redshift-S3-Write` è concesso a `group1` e `reg_user1`. Gli utenti in `group_1` possono specificare il ruolo solo per le operazioni COPY e l'utente `reg_user1` può specificare il ruolo solo per eseguire operazioni UNLOAD.

```
select pg_get_grantee_by_iam_role('arn:aws:iam::123456789012:role/Redshift-S3-Write');
```

```
  pg_get_grantee_by_iam_role
-----------------------------
 (group_1,g,COPY)
 (reg_user1,u,UNLOAD)
```

Nell'esempio seguente della funzione PG_GET_GRANTEE_BY_IAM_ROLE il risultato viene formattato come tabella.

```
select grantee, grantee_type, cmd_type FROM
 pg_get_grantee_by_iam_role('arn:aws:iam::123456789012:role/Redshift-S3-Write')
 res_grantee(grantee text, grantee_type text, cmd_type text) ORDER BY 1,2,3;
```

```
  grantee  | grantee_type | cmd_type
-----------+--------------+----------
 group_1   | g            | COPY
 reg_user1 | u            | UNLOAD
```

PG_GET_IAM_ROLE_BY_USER

Restituisce tutti i ruoli IAM e i privilegi di comando concessi a un utente.

Sintassi

```
pg_get_iam_role_by_user('name')
```

Argomenti

name

Il nome dell'utente per il quale restituire i ruoli IAM.

Tipo restituito

VARCHAR

Note per l'utilizzo

La funzione PG_GET_IAM_ROLE_BY_USER restituisce una riga per ogni set di ruoli e privilegi per il comando. La riga contiene un elenco separato da virgole con il nome utente, il ruolo IAM e il comando.

Un valore di `default` nel risultato indica che l'utente può specificare qualsiasi ruolo disponibile per eseguire il comando visualizzato.

Per utilizzare questa funzione, è necessario essere un utente con privilegi avanzati.

Esempio

L'esempio seguente indica che l'utente `reg_user1` può specificare qualsiasi ruolo IAM disponibile per eseguire operazioni COPY. L'utente può specificare il ruolo `Redshift-S3-Write` anche per le operazioni UNLOAD.

```
select pg_get_iam_role_by_user('reg_user1');
```

```
                            pg_get_iam_role_by_user
-------------------------------------------------------------------------------
 (reg_user1,default,COPY)
 (reg_user1,arn:aws:iam::123456789012:role/Redshift-S3-Write,COPY|UNLOAD)
```

Nell'esempio seguente della funzione PG_GET_IAM_ROLE_BY_USER il risultato viene formattato come tabella.

```
select username, iam_role, cmd FROM pg_get_iam_role_by_user('reg_user1')
 res_iam_role(username text, iam_role text, cmd text);
```

```
 username  |                    iam_role                     | cmd
-----------+-------------------------------------------------+------
 reg_user1 | default                                         | None
 reg_user1 | arn:aws:iam::123456789012:role/Redshift-S3-Read | COPY
```

PG_GET_LATE_BINDING_VIEW_COLS

Restituisce i metadati della colonna per tutte le visualizzazioni tardive nel database. Per ulteriori informazioni, consultare Viste con associazione tardiva

Sintassi

```
pg_get_late_binding_view_cols()
```

Tipo restituito

VARCHAR

Note per l'utilizzo

La funzione PG_GET_LATE_BINDING_VIEW_COLS restituisce una riga per ogni colonna nelle visualizzazioni tardive. La riga contiene un elenco separato da virgole con il nome dello schema, il nome della relazione, il nome della colonna, il tipo di dati e il numero di colonna.

Esempio

Il seguente esempio restituisce i metadati della colonna per tutte le visualizzazioni tardive.

```
select pg_get_late_binding_view_cols();

pg_get_late_binding_view_cols
------------------------------------------------------------
(public,myevent,eventname,"character varying(200)",1)
(public,sales_lbv,salesid,integer,1)
(public,sales_lbv,listid,integer,2)
(public,sales_lbv,sellerid,integer,3)
(public,sales_lbv,buyerid,integer,4)
(public,sales_lbv,eventid,integer,5)
```

```
(public,sales_lbv,dateid,smallint,6)
(public,sales_lbv,qtysold,smallint,7)
(public,sales_lbv,pricepaid,"numeric(8,2)",8)
(public,sales_lbv,commission,"numeric(8,2)",9)
(public,sales_lbv,saletime,"timestamp without time zone",10)
(public,event_lbv,eventid,integer,1)
(public,event_lbv,venueid,smallint,2)
(public,event_lbv,catid,smallint,3)
(public,event_lbv,dateid,smallint,4)
(public,event_lbv,eventname,"character varying(200)",5)
(public,event_lbv,starttime,"timestamp without time zone",6)
```

Il seguente esempio restituisce i metadati della colonna per tutte le visualizzazioni nel formato tabella.

```
select * from pg_get_late_binding_view_cols() cols(view_schema name, view_name name,
 col_name name, col_type varchar, col_num int);
view_schema | view_name | col_name   | col_type                    | col_num
------------+-----------+------------+-----------------------------+--------
public      | sales_lbv | salesid    | integer                     |       1
public      | sales_lbv | listid     | integer                     |       2
public      | sales_lbv | sellerid   | integer                     |       3
public      | sales_lbv | buyerid    | integer                     |       4
public      | sales_lbv | eventid    | integer                     |       5
public      | sales_lbv | dateid     | smallint                    |       6
public      | sales_lbv | qtysold    | smallint                    |       7
public      | sales_lbv | pricepaid  | numeric(8,2)                |       8
public      | sales_lbv | commission | numeric(8,2)                |       9
public      | sales_lbv | saletime   | timestamp without time zone |      10
public      | event_lbv | eventid    | integer                     |       1
public      | event_lbv | venueid    | smallint                    |       2
public      | event_lbv | catid      | smallint                    |       3
public      | event_lbv | dateid     | smallint                    |       4
public      | event_lbv | eventname  | character varying(200)      |       5
public      | event_lbv | starttime  | timestamp without time zone |       6
```

PG_GET_SESSION_ROLES

Restituisce i ruoli di sessione dell'utente attualmente connesso. I ruoli di sessione di un utente sono i gruppi definiti da un gestore dell'identità digitale per l'utente che ha effettuato l'accesso. Ad esempio, un gestore dell'identità digitale comeMicrosoft Azure Active Directory (Azure AD) verifica l'identità dell'utente e fornisce gli eventuali gruppi esterni di cui l'utente fa parte durante il processo di accesso dell'utente. Questi gruppi esterni vengono trasformati in ruoli Amazon Redshift e sono disponibili

durante la sessione corrente. Questi ruoli sono chiamati ruoli di sessione. Un amministratore può concedere privilegi a un ruolo di sessione simili ad altri ruoli di Amazon Redshift. Per ulteriori informazioni sull'uso dei ruoli, consulta Controllo accessi basato sui ruoli (RBAC). Per informazioni sulla gestione delle identità con un gestore dell'identità digitale, consulta Native identity provider (IdP) federation for Amazon Redshift (Federazione di provider di identità nativi (IdP) per Amazon Redshift) nella Guida alla gestione di Amazon Redshift.

Per visualizzare i ruoli definiti nel catalogo Amazon Redshift, connettiti al database come amministratore o come utente con privilegi avanzati ed esegui una query sulla vista di sistema SVV_ROLES.

Sintassi

```
pg_get_session_roles()
```

Tipo restituito

Un set di righe composto da due valori. Il primo valore è composto da due parti separate da due punti (:) che contiene un `idp-namespace:role-name`. `idp-namespace` è lo spazio dei nomi del gestore dell'identità digitale. `role-name` è il nome del gruppo esterno nel gestore dell'identità digitale. Il secondo valore contiene un `role-id` che è l'identificatore del ruolo.

Note per l'utilizzo

La funzione `PG_GET_SESSION_ROLES` restituisce una riga per ogni ruolo di sessione restituito.

Esempi

L'esempio seguente restituisce una riga per ogni ruolo dall'IdP di Azure Active Directory. Le colonne restituite vengono convertite in `sess_roles` con colonne `name` e `roleid`. Ciascun `name` è costituito dallo spazio dei nomi di Azure Active Directory e da un nome di gruppo in Azure Active Directory.

```
SELECT * FROM pg_get_session_roles() AS sess_roles(name name, roleid integer);

name                   roleid
--------------------------------
my_aad:test_group_1   106204
my_aad:test_group_2   106205
my_aad:test_group_3   106206
my_aad:test_group_4   106207
my_aad:test_group_5   106208
```

L'esempio seguente restituisce una riga per ogni gruppo IAM di cui l'utente IAM attualmente connesso è membro. Le colonne restituite vengono convertite in `sess_roles` con colonne `name` e `roleid`. Ciascun `name` è costituito dallo spazio dei nomi IAM e dal nome del gruppo IAM.

```
SELECT * FROM pg_get_session_roles() AS sess_roles(name name, roleid integer);

name                  roleid
-------------------------------
IAM:myGroup           110332
```

PG_LAST_COPY_COUNT

Restituisce il numero di righe che sono state caricate dall'ultimo comando COPY eseguito nella sessione corrente. PG_LAST_COPY_COUNT viene aggiornato con l'ultimo ID COPY, che è l'ID della query dell'ultima COPY che ha iniziato il processo di caricamento, anche se il caricamento non ha avuto esito positivo. L'ID query e l'ID COPY vengono aggiornati quando il comando COPY inizia il processo di caricamento.

Se il COPY ha esito negativo a causa di un errore di sintassi o di privilegi insufficienti, l'ID COPY non viene aggiornato e PG_LAST_COPY_COUNT restituisce il conteggio per la il COPY precedente. Se non sono stati eseguiti comandi COPY nella sessione corrente o se l'ultimo COPY ha avuto esito negativo durante il caricamento, PG_LAST_COPY_COUNT restituisce 0. Per ulteriori informazioni, consulta PG_LAST_COPY_ID.

Sintassi

```
pg_last_copy_count()
```

Tipo restituito

Restituisce BIGINT.

Esempio

La seguente query restituisce il numero di righe caricate dal comando COPY più recente eseguito nella sessione corrente.

```
select pg_last_copy_count();

pg_last_copy_count
--------------------
```

```
             192497
(1 row)
```

PG_LAST_COPY_ID

Restituisce l'ID della query del comando COPY completato più recentemente nella sessione corrente. Se non sono stati eseguiti comandi COPY nella sessione corrente, PG_LAST_COPY_ID restituisce -1.

Il valore per PG_LAST_COPY_ID viene aggiornato quando il comando COPY inizia il processo di caricamento. Se il COPY ha esito negativo a causa di dati di carico non validi, l'ID COPY viene aggiornato, quindi è possibile utilizzare PG_LAST_COPY_ID quando si esegue una query sulla tabella STL_LOAD_ERRORS. Se si esegue il rollback della transazione COPY, l'ID COPY non viene aggiornato.

L'ID COPY non viene aggiornato se il comando COPY ha esisto negativo a causa di un errore che si verifica prima dell'inizio del processo di caricamento, come ad esempio un errore di sintassi, un errore di accesso, credenziali non valide o privilegi insufficienti. L'ID COPY non viene aggiornato se COPY ha esito negativo durante la fase di compressione dell'analisi, che inizia dopo una connessione riuscita, ma prima del caricamento dei dati.

Sintassi

```
pg_last_copy_id()
```

Tipo restituito

Restituisce un integer.

Esempio

La seguente query restituisce l'ID della query dal comando COPY più recente eseguito nella sessione corrente.

```
select pg_last_copy_id();

pg_last_copy_id
---------------
          5437
(1 row)
```

La query seguente collega STL_LOAD_ERRORS a STL_LOADERROR_DETAIL per visualizzare gli errori dei dettagli che si sono verificati durante il caricamento più recente nella sessione corrente:

```
select d.query, substring(d.filename,14,20),
d.line_number as line,
substring(d.value,1,16) as value,
substring(le.err_reason,1,48) as err_reason
from stl_loaderror_detail d, stl_load_errors le
where d.query = le.query
and d.query = pg_last_copy_id();

 query |    substring      | line |  value   |                   err_reason
-------+-------------------+------+----------
+-----------------------------------------------
    558| allusers_pipe.txt |  251 | 251      | String contains invalid or unsupported
 UTF8 code
    558| allusers_pipe.txt |  251 | ZRU29FGR | String contains invalid or unsupported
 UTF8 code
    558| allusers_pipe.txt |  251 | Kaitlin  | String contains invalid or unsupported
 UTF8 code
    558| allusers_pipe.txt |  251 | Walter   | String contains invalid or unsupported
 UTF8 code
```

PG_LAST_UNLOAD_ID

Restituisce l'ID della query del comando UNLOAD completato più recentemente nella sessione corrente. Se non sono stati eseguiti comandi UNLOAD nella sessione corrente, PG_LAST_UNLOAD_ID restituisce -1.

Il valore per PG_LAST_UNLOAD_ID viene aggiornato quando il comando UNLOAD inizia il processo di caricamento. Se il UNLOAD ha esito negativo a causa di dati di carico non validi, l'ID UNLOAD viene aggiornato, quindi è possibile utilizzare l'ID UNLOAD per ulteriori ricerche. Se si esegue il rollback della transazione UNLOAD, l'ID UNLOAD non viene aggiornato.

L'ID UNLOAD non viene aggiornato se il comando UNLOAD ha esisto negativo a causa di un errore che si verifica prima dell'inizio del processo di caricamento, come ad esempio un errore di sintassi, un errore di accesso, credenziali non valide o privilegi insufficienti.

Sintassi

```
PG_LAST_UNLOAD_ID()
```

Tipo restituito

Restituisce un integer.

Esempio

La seguente query restituisce l'ID della query dal comando UNLOAD più recente eseguito nella sessione corrente.

```
select PG_LAST_UNLOAD_ID();

PG_LAST_UNLOAD_ID
---------------
          5437
(1 row)
```

PG_LAST_QUERY_ID

Restituisce l'ID della query completata più recentemente nella sessione corrente. Se non sono state eseguite query nella sessione corrente, PG_LAST_QUERY_ID restituisce -1. PG_LAST_QUERY_ID non restituisce l'ID query per le query eseguite esclusivamente sul nodo principale. Per ulteriori informazioni, consulta Nodo principale: solo funzioni.

Sintassi

```
pg_last_query_id()
```

Tipo restituito

Restituisce un integer.

Esempio

La seguente query restituisce l'ID dell'ultima query completata nella sessione corrente.

```
select pg_last_query_id();
```

Di seguito sono riportati i risultati.

```
pg_last_query_id
----------------
           5437
```

```
(1 row)
```

La query seguente restituisce l'ID di query e il testo della query completata più di recente nella sessione corrente.

```
select query, trim(querytxt) as sqlquery
from stl_query
where query = pg_last_query_id();
```

Di seguito sono riportati i risultati.

```
query | sqlquery
------+-------------------------------------------------
 5437 | select name, loadtime from stl_file_scan where loadtime > 1000000;
(1 rows)
```

PG_LAST_UNLOAD_COUNT

Restituisce il numero di righe che non sono state caricate dall'ultimo comando UNLOAD completato nella sessione corrente. PG_LAST_UNLOAD_COUNT viene aggiornato con l'ultimo ID query dell'ultimo UNLOAD, anche se l'operazione non ha avuto esito positivo. L'ID query viene aggiornato quando viene completato l'UNLOAD. Se l'UNLOAD ha esito negativo a causa di un errore di sintassi o di privilegi insufficienti, PG_LAST_UNLOAD_COUNT restituisce il conteggio per la l'UNLOAD precedente. Se non sono stati completati comandi UNLOAD nella sessione corrente o se l'ultimo UNLOAD ha avuto esito negativo durante l'operazione di caricamento, PG_LAST_UNLOAD_COUNT restituisce 0.

Sintassi

```
pg_last_unload_count()
```

Tipo restituito

Restituisce BIGINT.

Esempio

La seguente query restituisce il numero di righe scaricate dal comando UNLOAD più recente eseguito nella sessione corrente.

```
select pg_last_unload_count();
```

```
pg_last_unload_count
--------------------
              192497
(1 row)
```

SLICE_NUM Function

Restituisce un integer corrispondente al numero di sezione nel cluster in cui si trovano i dati di una riga. SLICE_NUM non assume parametri.

Sintassi

```
SLICE_NUM()
```

Tipo restituito

La funzione SLICE_NUM restituisce un integer.

Esempi

L'esempio seguente mostra quali sezioni contengono dati per le prime dieci righe EVENT nella tabella EVENTS:

```
select distinct eventid, slice_num() from event order by eventid limit 10;

 eventid | slice_num
---------+-----------
       1 |         1
       2 |         2
       3 |         3
       4 |         0
       5 |         1
       6 |         2
       7 |         3
       8 |         0
       9 |         1
      10 |         2
(10 rows)
```

L'esempio seguente restituisce un codice (10000) per mostrare che una query senza un'istruzione FROM viene eseguita sul nodo principale:

```
select slice_num();
slice_num
-----------
10000
(1 row)
```

UTENTE

Sinonimo di CURRENT_USER. Per informazioni, consultare CURRENT_USER.

ROLE_IS_MEMBER_OF

Restituisce true se il ruolo è membro di un altro ruolo. Gli utenti con privilegi avanzati possono controllare l'appartenenza di tutti i ruoli. Gli utenti abituali che dispongono dell'autorizzazione ACCESS SYSTEM TABLE possono controllare l'appartenenza di tutti gli utenti. In caso contrario, gli utenti normali controllano solo i ruoli a cui hanno accesso. Amazon Redshift restituisce un errore se i ruoli forniti non esistono o se l'utente corrente non ha accesso al ruolo.

Sintassi

```
role_is_member_of( role_name,  granted_role_name)
```

Argomenti

role_name

Il nome del ruolo.

granted_role_name

Il nome del ruolo concesso.

Tipo restituito

Restituisce un BOOLEAN.

Esempio

La seguente query conferma che il ruolo non è membro di role1 né di role2.

```
SELECT role_is_member_of('role1', 'role2');
```

```
 role_is_member_of
-------------------
             False
```

USER_IS_MEMBER_OF

Restituisce true se l'utente è un membro di un ruolo o di un gruppo. Gli utenti con privilegi avanzati possono controllare l'appartenenza di tutti gli utenti. Gli utenti normali che sono membri del ruolo sys:secadmin o sys:superuser possono controllare l'appartenenza di tutti gli utenti. In caso contrario, gli utenti normali possono solo controllare loro stessi. Amazon Redshift restituisce un errore se le identità fornite non esistono o se l'utente corrente non ha accesso al ruolo.

Sintassi

```
user_is_member_of( user_name,  role_name | group_name)
```

Argomenti

user_name

Il nome dell'utente.

role_name

Il nome del ruolo.

group_name

Il nome del gruppo.

Tipo restituito

Restituisce un BOOLEAN.

Esempio

La seguente query conferma che l'utente non è membro di role1.

```
SELECT user_is_member_of('reguser', 'role1');

 user_is_member_of
-------------------
           False
```

VERSION

La funzione VERSION restituisce i dettagli della versione correntemente installata, con informazioni sulla versione di Amazon Redshift specifiche alla fine.

> **Note**
>
> Questa è una funzione del nodo principale. Questa funzione restituisce un errore se fa riferimento a una tabella creata dall'utente, a una tabella di sistema STL o STV o a una vista di sistema SVV o SVL.

Sintassi

```
VERSION()
```

Tipo restituito

Restituisce una stringa CHAR o VARCHAR.

Esempi

Nell'esempio seguente vengono illustrate le informazioni sulla versione del cluster corrente:

```
select version();
```

```
 version
----------------------------------------------------------------------------------------------------

PostgreSQL 8.0.2 on i686-pc-linux-gnu, compiled by GCC gcc (GCC) 3.4.2 20041017 (Red
Hat 3.4.2-6.fc3), Redshift 1.0.12103
```

Dove `1.0.12103` è il numero di versione del cluster.

> **Note**
>
> Per forzare l'aggiornamento del cluster alla versione più recente, modifica la finestra di manutenzione.

Parole riservate

Di seguito è riportato un elenco delle parole riservate di Amazon Redshift. È possibile usare le parole riservate con identificatori delimitati (virgolette doppie).

> **Note**
>
> Sebbene START e CONNECT non siano parole riservate, utilizzare identificatori delimitati o AS se si utilizza START e CONNECT come alias di tabella nella query per evitare errori in fase di runtime.

Per ulteriori informazioni, consulta Nomi e identificatori.

```
AES128
AES256
ALL
ALLOWOVERWRITE
ANALYSE
ANALYZE
AND
ANY
ARRAY
AS
ASC
AUTHORIZATION
AZ64
BACKUP
BETWEEN
BINARY
BLANKSASNULL
BOTH
BYTEDICT
BZIP2
CASE
CAST
CHECK
COLLATE
COLUMN
CONSTRAINT
CREATE
CREDENTIALS
```

```
CROSS
CURRENT_DATE
CURRENT_TIME
CURRENT_TIMESTAMP
CURRENT_USER
CURRENT_USER_ID
DEFAULT
DEFERRABLE
DEFLATE
DEFRAG
DELTA
DELTA32K
DESC
DISABLE
DISTINCT
DO
ELSE
EMPTYASNULL
ENABLE
ENCODE
ENCRYPT
ENCRYPTION
END
EXCEPT
EXPLICIT
FALSE
FOR
FOREIGN
FREEZE
FROM
FULL
GLOBALDICT256
GLOBALDICT64K
GRANT
GROUP
GZIP
HAVING
IDENTITY
IGNORE
ILIKE
IN
INITIALLY
INNER
INTERSECT
```

```
INTERVAL
INTO
IS
ISNULL
JOIN
LEADING
LEFT
LIKE
LIMIT
LOCALTIME
LOCALTIMESTAMP
LUN
LUNS
LZO
LZOP
MINUS
MOSTLY16
MOSTLY32
MOSTLY8
NATURAL
NEW
NOT
NOTNULL
NULL
NULLS
OFF
OFFLINE
OFFSET
OID
OLD
ON
ONLY
OPEN
OR
ORDER
OUTER
OVERLAPS
PARALLEL
PARTITION
PERCENT
PERMISSIONS
PIVOT
PLACING
PRIMARY
```

```
RAW
READRATIO
RECOVER
REFERENCES
REJECTLOG
RESORT
RESPECT
RESTORE
RIGHT
SELECT
SESSION_USER
SIMILAR
SNAPSHOT
SOME
SYSDATE
SYSTEM
TABLE
TAG
TDES
TEXT255
TEXT32K
THEN
TIMESTAMP
TO
TOP
TRAILING
TRUE
TRUNCATECOLUMNS
UNION
UNIQUE
UNNEST
UNPIVOT
USER
USING
VERBOSE
WALLET
WHEN
WHERE
WITH
WITHOUT
```

Riferimento di tabelle e viste di sistema

Argomenti

Tabelle e viste di sistema

Amazon Redshift include numerose tabelle e viste di sistema che contengono informazioni sul funzionamento del sistema. Puoi eseguire delle query su queste tabelle e viste esattamente come faresti con altre tabelle di database. Questa sezione include alcuni esempi di query di tabella di sistema e descrive:

- Il modo in cui differenti tipi di tabelle e viste di sistema sono generati
- Quali tipi di informazioni è possibile ottenere da queste tabelle
- Come eseguire il join di tabelle di sistema di Amazon Redshift a tabelle di catalogo
- Come gestire l'aumento delle dimensioni dei file di log relativi alle tabelle di sistema

Alcune tabelle di sistema possono essere utilizzate dal AWS personale solo per scopi diagnostici. Le sezioni seguenti descrivono le tabelle di sistema che gli amministratori di sistema o altri utenti di database possono sottoporre a query allo scopo di ottenere informazioni utili.

> **Note**
>
> Le tabelle di sistema non sono incluse nei backup automatici o manuali dei cluster (snapshot). Le visualizzazioni di sistema STL mantengono per sette giorni la cronologia dei log. La conservazione dei log non richiede alcuna azione da parte del cliente, ma se desideri archiviare i dati di log per più di 7 giorni, devi copiarli periodicamente in altre tabelle o scaricarli in Amazon S3.

Tipi di tabelle e viste di sistema

Esistono diversi tipi di tabelle e viste di sistema:

- Le viste SVV contengono informazioni sugli oggetti del database con riferimenti a tabelle STV transitorie.
- Viste SYS sono utilizzate per monitorare l'utilizzo di query e carichi di lavoro dei cluster con provisioning e dei gruppi di lavoro serverless.
- Le viste STL sono generate a partire dai log conservati su disco per fornire una cronologia del sistema.
- Le tabelle STV sono tabelle di sistema virtuali che contengono snapshot dei dati di sistema correnti. Sono basate su dati in memoria transitori e non sono conservate in tabelle normali o log basati su disco.
- Le viste SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento simultaneo.
- Le viste SVL forniscono dettagli sulle query sui cluster principali.

Le tabelle e le viste di sistema non utilizzano lo stesso modello di consistenza delle tabelle normali. È importante essere consapevoli di questo aspetto quando si effettuano delle query sulle stesse, soprattutto per le tabelle STV e le viste SVV. Ad esempio, nel caso di una tabella normale t1 con una colonna c1, la query seguente non dovrebbe restituire righe:

```
select * from t1
where c1 > (select max(c1) from t1)
```

Tuttavia, la query seguente su una tabella di sistema potrebbe restituire delle righe:

```
select * from stv_exec_state
where currenttime > (select max(currenttime) from stv_exec_state)
```

Il motivo è che currentime è transitorio e i due riferimenti nella query potrebbero non restituire lo stesso valore quando valutati.

D'altra parte, la query seguente potrebbe non restituire delle righe:

```
select * from stv_exec_state
where currenttime = (select max(currenttime) from stv_exec_state)
```

Visibilità dei dati nelle tabelle e nelle viste di sistema

Vi sono due classi di visibilità dei dati nelle tabelle e nelle viste di sistema: visibili agli utenti e visibili agli utenti con privilegi avanzati.

Soltanto gli utenti con privilegi avanzati possono vedere i dati nelle tabelle della categoria visibile agli utenti con privilegi avanzati. Gli utenti standard possono vedere i dati nella tabelle visibili agli utenti. Per consentire a un utente regolare l'accesso alle tabelle visibili da superutente, concedi il privilegio SELECT su quella tabella all'utente normale. Per ulteriori informazioni, consulta GRANT.

Per impostazione predefinita, nella maggior parte delle tabelle visibili agli utenti, le righe generate da un altro utente sono invisibili a un utente standard. Se a un utente normale viene concesso SYSLOG ACCESS UNRESTRICTED, quell'utente può vedere tutte le righe nelle tabelle visibili all'utente, comprese le righe generate da un altro utente. Per ulteriori informazioni, consultare ALTER USER o CREA UTENTE. Tutte le righe in SVV_TRANSACTIONS sono visibili a tutti gli utenti. Per ulteriori informazioni sulla visibilità dei dati, consulta l'articolo della AWS re:Post knowledge base Come posso consentire agli utenti regolari del database Amazon Redshift l'autorizzazione a visualizzare i dati nelle tabelle di sistema di altri utenti del mio cluster? .

Per quanto riguarda le visualizzazioni dei metadati, Amazon Redshift non consente la visibilità agli utenti a cui è concesso SYSLOG ACCESS UNRESTRICTED.

> **Note**
>
> Concedere all'utente un accesso illimitato alle tabelle di sistema offre all'utente la visibilità dei dati generati da altri utenti. Ad esempio, STL_QUERY e STL_QUERY_TEXT contengono il

testo completo delle istruzioni INSERT, UPDATE e DELETE, che potrebbero contenere dati sensibili generati dall'utente.

un utente con privilegi avanzati può vedere tutte le righe in tutte le tabelle. Per consentire a un utente standard di accedere alle tabelle visibili all'utente con privilegi avanzati, è necessario concedergli (GRANT) il privilegio SELECT su quelle tabelle.

Filtraggio delle query generate dal sistema

Le tabelle e le viste di sistema correlate alle query, come SVL_QUERY_SUMMARY, SVL_QLOG e altre, in genere contengono un gran numero di istruzioni generate automaticamente che Amazon Redshift utilizza per monitorare lo stato del database. Queste query generate dal sistema sono visibili a un utente con privilegi avanzati, ma sono raramente utili. Per filtrarle quando si esegue una selezione da una tabella o da una vista di sistema che utilizza la colonna `userid`, è necessario aggiungere la condizione `userid > 1` alla clausola WHERE. Per esempio:

```
select * from svl_query_summary where userid > 1
```

Migrazione di query solo predisposte a query di visualizzazione di monitoraggio SYS

Migrazione dai cluster con provisioning ad Amazon Redshift serverless

Se stai migrando un cluster con provisioning su Amazon Redshift Serverless, potresti avere domande utilizzando le seguenti viste di sistema, che memorizzano solo i dati dei cluster forniti.

- Tutte le viste STL
- Tutte le viste STV
- Tutte le viste SVCS
- Tutte le viste SVL
- Alcune viste SVV
  - Per un elenco completo delle viste SVV non supportate in Amazon Redshift Serverless, consulta l'elenco in fondo alla pagina Monitoraggio delle query e dei carichi di lavoro con Amazon Redshift Serverless nella Amazon Redshift Management Guide.

Per continuare a utilizzare le tue query, modificale in modo da utilizzare le colonne definite nelle viste di monitoraggio SYS che corrispondono alle colonne delle tue viste solo con provisioning. Per vedere la relazione di mappatura tra le viste di solo provisioning e le viste di monitoraggio SYS, vai a Mappatura delle viste di sistema per la migrazione alle viste di monitoraggio SYS

Aggiornamento delle query restando in un cluster con provisioning

Se non stai effettuando la migrazione ad Amazon Redshift serverless, potresti comunque voler aggiornare le query esistenti. Le viste di monitoraggio SYS sono progettate per facilitare l'uso e ridurre la complessità, fornendo una gamma completa di parametri per un monitoraggio e una risoluzione dei problemi efficaci. Utilizzando viste SYS come SYS_QUERY_HISTORY e SYS_QUERY_DETAIL che consolidano le informazioni di più viste solo con provisioning, è possibile semplificare le query.

Miglioramento del monitoraggio degli identificatori di query utilizzando le viste di monitoraggio SYS

Le viste di monitoraggio SYS come ad esempio SYS_QUERY_HISTORY e SYS_QUERY_DETAIL contengono la colonna query_id con l'identificatore delle query degli utenti. Analogamente, anche le viste solo con provisioning, come STL_QUERY e SVL_QLOG, contengono la colonna di query con gli identificatori delle query. Tuttavia, gli identificatori di query registrati nelle viste del sistema SYS sono diversi da quelli registrati nelle viste solo con provisioning.

La differenza tra i valori delle colonne query_id delle viste SYS e i valori delle colonne di query delle viste solo con provisioning è la seguente:

- Nelle viste SYS, la colonna query_id registra le query inviate dall'utente nella forma originale. Il sistema di ottimizzazione di Amazon Redshift può suddividerle in query figlio per migliorare le prestazioni, ma ogni singola query eseguita avrà comunque una propria riga in SYS_QUERY_HISTORY. Se desideri visualizzare le singole query figlio, sono disponibili in SYS_QUERY_DETAIL.
- Nelle viste solo con provisioning, la colonna di query registra le query a livello di query figlio. Se il sistema di ottimizzazione di Amazon Redshift riscrive la query originale in più query figlio, verranno inserite più righe in STL_QUERY con identificatori di query diversi per ogni singola query eseguita.

Quando esegui la migrazione delle query di monitoraggio e diagnostica da viste solo con provisioning a viste SYS, considera questa differenza e modifica le query di conseguenza. Per ulteriori

informazioni su come Amazon Redshift elabora le query, consulta Pianificazione di query e flusso di lavoro di esecuzione.

Esempio

Per un esempio di come Amazon Redshift registra le query in modo diverso nelle viste di monitoraggio solo provisioning e SYS, consulta la seguente query di esempio. Questa query è scritta per essere eseguita in Amazon Redshift.

```
SELECT
  s_name
  , COUNT(*) AS numwait
FROM
  supplier,
  lineitem l1,
  orders,
  nation
WHERE    s_suppkey = l1.l_suppkey
         AND o_orderkey = l1.l_orderkey
         AND o_orderstatus = 'F'
         AND l1.l_receiptdate > l1.l_commitdate
         AND EXISTS (SELECT
                       *
                     FROM
                       lineitem l2
                     WHERE  l2.l_orderkey = l1.l_orderkey
                            AND l2.l_suppkey <> l1.l_suppkey )
         AND NOT EXISTS (SELECT
                           *
                         FROM
                           lineitem l3
                         WHERE  l3.l_orderkey = l1.l_orderkey
                                AND l3.l_suppkey <> l1.l_suppkey
                                AND l3.l_receiptdate > l3.l_commitdate )
         AND s_nationkey = n_nationkey
         AND n_name = 'UNITED STATES'
GROUP BY
  s_name
ORDER BY
  numwait DESC
  , s_name LIMIT 100;
```

Dietro le quinte, il sistema di ottimizzazione di query di Amazon Redshift riscrive la precedente query inviata dall'utente in cinque query figlio.

La prima query figlio crea una tabella temporanea per materializzare una sottoquery.

```
CREATE TEMP TABLE volt_tt_606590308b512(l_orderkey
                                       , l_suppkey
                                       , s_name   ) AS SELECT
                                                         l1.l_orderkey
                                                         , l1.l_suppkey
                                                         , public.supplier.s_name
                                                       FROM
                                                         public.lineitem AS l1,
                                                         public.nation,
                                                         public.orders,
                                                         public.supplier
                                                       WHERE  l1.l_commitdate <
 l1.l_receiptdate
                                                              AND l1.l_orderkey =
 public.orders.o_orderkey
                                                              AND l1.l_suppkey =
 public.supplier.s_suppkey
                                                              AND public.nation.n_name
 = 'UNITED STATES'::CHAR(8)
                                                              AND
 public.nation.n_nationkey = public.supplier.s_nationkey
                                                              AND
 public.orders.o_orderstatus = 'F'::CHAR(1);
```

La seconda query figlio raccoglie le statistiche dalla tabella temporanea.

```
padb_fetch_sample: select count(*) from volt_tt_606590308b512;
```

La terza query figlio crea un'altra tabella temporanea per materializzare un'altra sottoquery, facendo riferimento alla tabella temporanea creata in precedenza.

```
CREATE TEMP TABLE volt_tt_606590308c2ef(l_orderkey
                                       , l_suppkey) AS (SELECT

 volt_tt_606590308b512.l_orderkey
                                                          ,
 volt_tt_606590308b512.l_suppkey
```

```
                                                    FROM
                                                      public.lineitem AS l2,
                                                      volt_tt_606590308b512
                                                    WHERE  l2.l_suppkey <>
 volt_tt_606590308b512.l_suppkey
                                                           AND l2.l_orderkey =
 volt_tt_606590308b512.l_orderkey)
                                                           EXCEPT distinct (SELECT
 volt_tt_606590308b512.l_orderkey, volt_tt_606590308b512.l_suppkey
                                                           FROM public.lineitem AS
 l3, volt_tt_606590308b512
                                                           WHERE l3.l_commitdate <
 l3.l_receiptdate
                                                             AND l3.l_suppkey <>
 volt_tt_606590308b512.l_suppkey
                                                             AND l3.l_orderkey =
 volt_tt_606590308b512.l_orderkey);
```

La quarta query figlio raccoglie nuovamente le statistiche dalla tabella temporanea.

```
padb_fetch_sample: select count(*) from volt_tt_606590308c2ef
```

L'ultima query figlio utilizza le tabelle temporanee create in precedenza per generare l'output.

```
SELECT
  volt_tt_606590308b512.s_name AS s_name
  , COUNT(*) AS numwait
FROM
  volt_tt_606590308b512,
  volt_tt_606590308c2ef
WHERE    volt_tt_606590308b512.l_orderkey = volt_tt_606590308c2ef.l_orderkey
         AND volt_tt_606590308b512.l_suppkey = volt_tt_606590308c2ef.l_suppkey
GROUP BY
  1
ORDER BY
  2 DESC
  , 1 ASC LIMIT 100;
```

Nella vista di sistema solo con provisioning STL_QUERY, Amazon Redshift registra cinque righe a livello di query figlio, come segue:

```
SELECT userid, xid, pid, query, querytxt::varchar(100);
```

```
FROM stl_query
WHERE xid = 48237350
ORDER BY xid, starttime;

 userid |   xid    |    pid     |  query   |
    querytxt
--------+----------+------------+----------
+-----------------------------------------------------------------------------------------------------
    101 | 48237350 | 1073840810 | 12058151 | CREATE TEMP TABLE
 volt_tt_606590308b512(l_orderkey, l_suppkey, s_name) AS SELECT l1.l_orderkey, l1.l
    101 | 48237350 | 1073840810 | 12058152 | padb_fetch_sample: select count(*) from
 volt_tt_606590308b512
    101 | 48237350 | 1073840810 | 12058156 | CREATE TEMP TABLE
 volt_tt_606590308c2ef(l_orderkey, l_suppkey) AS (SELECT volt_tt_606590308b512.l_or
    101 | 48237350 | 1073840810 | 12058168 | padb_fetch_sample: select count(*) from
 volt_tt_606590308c2ef
    101 | 48237350 | 1073840810 | 12058170 | SELECT s_name , COUNT(*) AS numwait FROM
 supplier, lineitem l1, orders, nation WHERE s_suppkey = l1.
(5 rows)
```

Nella vista di monitoraggio SYS SYS_QUERY_HISTORY, Amazon Redshift registra la query come segue:

```
SELECT user_id, transaction_id, session_id, query_id, query_text::varchar(100)
FROM sys_query_history
WHERE transaction_id = 48237350
ORDER BY start_time;

 user_id | transaction_id | session_id | query_id |
          query_text
---------+----------------+------------+----------
+-----------------------------------------------------------------------------------------------------
     101 |       48237350 | 1073840810 | 12058149 | SELECT s_name , COUNT(*) AS numwait
 FROM supplier, lineitem l1, orders, nation WHERE s_suppkey = l1.
```

In SYS_QUERY_DETAIL sono disponibili i dettagli a livello di query figlio che puoi filtrare usando il valore query_id di SYS_QUERY_HISTORY. La colonna child_query_sequence mostra l'ordine in cui vengono eseguite le query figlio. Per ulteriori informazioni sulle colonne in SYS_QUERY_DETAIL, consulta SYS_QUERY_DETAIL.

```
select user_id,
       query_id,
```

```
       child_query_sequence,
       stream_id,
       segment_id,
       step_id,
       start_time,
       end_time,
       duration,
       blocks_read,
       blocks_write,
       local_read_io,
       remote_read_io,
       data_skewness,
       time_skewness,
       is_active,
       spilled_block_local_disk,
       spilled_block_remote_disk
from sys_query_detail
where query_id = 12058149
      and step_id = -1
order by query_id,
         child_query_sequence,
         stream_id,
         segment_id,
         step_id;

 user_id | query_id | child_query_sequence | stream_id | segment_id | step_id |
       start_time         |          end_time          | duration | blocks_read |
 blocks_write | local_read_io | remote_read_io | data_skewness | time_skewness |
 is_active | spilled_block_local_disk | spilled_block_remote_disk
---------+----------+----------------------+-----------+------------+---------
+---------------------------+---------------------------+----------+-------------
+--------------+--------------+----------------+---------------+--------------
+-----------+--------------------------+---------------------------
     101 | 12058149 |                    1 |         0 |          0 |      -1 |
 2023-09-27 15:40:38.512415 | 2023-09-27 15:40:38.533333 |    20918 |           0 |
         0 |             0 |              0 |             0 |            44 | f
 |                        0 |                           0
     101 | 12058149 |                    1 |         1 |          1 |      -1 |
 2023-09-27 15:40:39.931437 | 2023-09-27 15:40:39.972826 |    41389 |          12 |
         0 |            12 |              0 |             0 |            77 | f
 |                        0 |                           0
     101 | 12058149 |                    1 |         2 |          2 |      -1 |
 2023-09-27 15:40:40.584412 | 2023-09-27 15:40:40.613982 |    29570 |          32 |
```

```
        0 |            32 |              0 |             0 |           25 | f
|                        0 |                           0
    101 | 12058149 |                    1 |         2 |          3 |      -1 |
2023-09-27 15:40:40.582038 | 2023-09-27 15:40:40.615758 |    33720 |           0 |
        0 |             0 |              0 |             0 |            1 | f
|                        0 |                           0
    101 | 12058149 |                    1 |         3 |          4 |      -1 |
2023-09-27 15:40:46.668766 | 2023-09-27 15:40:46.705456 |    36690 |          24 |
        0 |            15 |              0 |             0 |           17 | f
|                        0 |                           0
    101 | 12058149 |                    1 |         4 |          5 |      -1 |
2023-09-27 15:40:46.707209 | 2023-09-27 15:40:46.709176 |     1967 |           0 |
        0 |             0 |              0 |             0 |           18 | f
|                        0 |                           0
    101 | 12058149 |                    1 |         4 |          6 |      -1 |
2023-09-27 15:40:46.70656  | 2023-09-27 15:40:46.71289  |     6330 |           0 |
        0 |             0 |              0 |             0 |            0 | f
|                        0 |                           0
    101 | 12058149 |                    1 |         5 |          7 |      -1 |
2023-09-27 15:40:46.71405  | 2023-09-27 15:40:46.714343 |      293 |           0 |
        0 |             0 |              0 |             0 |            0 | f
|                        0 |                           0
    101 | 12058149 |                    2 |         0 |          0 |      -1 |
2023-09-27 15:40:52.083907 | 2023-09-27 15:40:52.087854 |     3947 |           0 |
        0 |             0 |              0 |             0 |           35 | f
|                        0 |                           0
    101 | 12058149 |                    2 |         1 |          1 |      -1 |
2023-09-27 15:40:52.089632 | 2023-09-27 15:40:52.091129 |     1497 |           0 |
        0 |             0 |              0 |             0 |           11 | f
|                        0 |                           0
    101 | 12058149 |                    2 |         1 |          2 |      -1 |
2023-09-27 15:40:52.089008 | 2023-09-27 15:40:52.091306 |     2298 |           0 |
        0 |             0 |              0 |             0 |            0 | f
|                        0 |                           0
    101 | 12058149 |                    3 |         0 |          0 |      -1 |
2023-09-27 15:40:56.882013 | 2023-09-27 15:40:56.897282 |    15269 |           0 |
        0 |             0 |              0 |             0 |           29 | f
|                        0 |                           0
    101 | 12058149 |                    3 |         1 |          1 |      -1 |
2023-09-27 15:40:59.718554 | 2023-09-27 15:40:59.722789 |     4235 |           0 |
        0 |             0 |              0 |             0 |           13 | f
|                        0 |                           0
    101 | 12058149 |                    3 |         2 |          2 |      -1 |
2023-09-27 15:40:59.800382 | 2023-09-27 15:40:59.807388 |     7006 |           0 |
```

```
        0 |             0 |              0 |             0 |           58 | f
|                        0 |                           0
    101 | 12058149 |                    3 |         3 |          3 |      -1 |
2023-09-27 15:41:06.488685 | 2023-09-27 15:41:06.493825 |     5140 |           0 |
        0 |             0 |              0 |             0 |           56 | f
|                        0 |                           0
    101 | 12058149 |                    3 |         3 |          4 |      -1 |
2023-09-27 15:41:06.486206 | 2023-09-27 15:41:06.497756 |    11550 |           0 |
        0 |             0 |              0 |             0 |            2 | f
|                        0 |                           0
    101 | 12058149 |                    3 |         4 |          5 |      -1 |
2023-09-27 15:41:06.499201 | 2023-09-27 15:41:06.500851 |     1650 |           0 |
        0 |             0 |              0 |             0 |           15 | f
|                        0 |                           0
    101 | 12058149 |                    3 |         4 |          6 |      -1 |
2023-09-27 15:41:06.498609 | 2023-09-27 15:41:06.500949 |     2340 |           0 |
        0 |             0 |              0 |             0 |            0 | f
|                        0 |                           0
    101 | 12058149 |                    3 |         5 |          7 |      -1 |
2023-09-27 15:41:06.502945 | 2023-09-27 15:41:06.503282 |      337 |           0 |
        0 |             0 |              0 |             0 |            0 | f
|                        0 |                           0
    101 | 12058149 |                    4 |         0 |          0 |      -1 |
2023-09-27 15:41:06.62899  | 2023-09-27 15:41:06.631452 |     2462 |           0 |
        0 |             0 |              0 |             0 |           22 | f
|                        0 |                           0
    101 | 12058149 |                    4 |         1 |          1 |      -1 |
2023-09-27 15:41:06.632313 | 2023-09-27 15:41:06.63391  |     1597 |           0 |
        0 |             0 |              0 |             0 |           20 | f
|                        0 |                           0
    101 | 12058149 |                    4 |         1 |          2 |      -1 |
2023-09-27 15:41:06.631726 | 2023-09-27 15:41:06.633813 |     2087 |           0 |
        0 |             0 |              0 |             0 |            0 | f
|                        0 |                           0
    101 | 12058149 |                    5 |         0 |          0 |      -1 |
2023-09-27 15:41:12.571974 | 2023-09-27 15:41:12.584234 |    12260 |           0 |
        0 |             0 |              0 |             0 |           39 | f
|                        0 |                           0
    101 | 12058149 |                    5 |         0 |          1 |      -1 |
2023-09-27 15:41:12.569815 | 2023-09-27 15:41:12.585391 |    15576 |           0 |
        0 |             0 |              0 |             0 |            4 | f
|                        0 |                           0
    101 | 12058149 |                    5 |         1 |          2 |      -1 |
2023-09-27 15:41:13.758513 | 2023-09-27 15:41:13.76401  |     5497 |           0 |
```

```
         0 |             0 |              0 |             0 |           39 | f
 |                        0 |                           0
     101 | 12058149 |                    5 |         1 |          3 |      -1 |
 2023-09-27 15:41:13.749    | 2023-09-27 15:41:13.772987 |    23987 |           0 |
         0 |             0 |              0 |             0 |           32 | f
 |                        0 |                           0
     101 | 12058149 |                    5 |         2 |          4 |      -1 |
 2023-09-27 15:41:13.799526 | 2023-09-27 15:41:13.813506 |    13980 |           0 |
         0 |             0 |              0 |             0 |           62 | f
 |                        0 |                           0
     101 | 12058149 |                    5 |         2 |          5 |      -1 |
 2023-09-27 15:41:13.798823 | 2023-09-27 15:41:13.813651 |    14828 |           0 |
         0 |             0 |              0 |             0 |            0 | f
 |                        0 |                           0
(28 rows)
```

ID di interrogazione, processo e sessione della tabella di sistema

Quando analizzate gli ID di interrogazione, processo e sessione visualizzati nelle tabelle di sistema, tenete presente quanto segue:

- Il valore dell'ID della query (in colonne come `query_id` `equery`) può essere riutilizzato nel tempo.
- L'id del processo o il valore dell'id della sessione (in colonne come `process_id`pid, `esession_id`) può essere riutilizzato nel tempo.
- Il valore dell'ID della transazione (in colonne come `transaction_id` `exid`) è unico.

Viste SVV dei metadati

Le viste SVV sono viste di sistema in Amazon Redshift che contengono informazioni sugli oggetti del database.

> **Note**
>
> Amazon Redshift segnala un WARNING, non un ERROR, se una risposta del database non riesce per qualsiasi motivo. Amazon Redshift non invia messaggi di tipo ERROR quando si stanno eseguendo query su oggetti in una unità di condivisione dati.

Argomenti

- SVV_ACTIVE_CURSORS
- SVV_ALL_COLUMNS
- SVV_ALL_SCHEMAS
- SVV_ALL_TABLES
- SVV_ALTER_TABLE_RECOMMENDATIONS
- SVV_ATTACHED_MASKING_POLICY
- SVV_COLUMNS
- SVV_COLUMN_PRIVILEGES
- SVV_DATABASE_PRIVILEGES
- SVV_DATASHARE_PRIVILEGES
- SVV_DATASHARES
- SVV_DATASHARE_CONSUMERS
- SVV_DATASHARE_OBJECTS
- SVV_DEFAULT_PRIVILEGES
- SVV_DISKUSAGE
- SVV_EXTERNAL_COLUMNS
- SVV_EXTERNAL_DATABASES
- SVV_EXTERNAL_PARTITIONS
- SVV_EXTERNAL_SCHEMAS
- SVV_EXTERNAL_TABLES
- SVV_FUNCTION_PRIVILEGES
- SVV_GEOGRAPHY_COLUMNS
- SVV_GEOMETRY_COLUMNS
- SVV_IAM_PRIVILEGES
- SVV_IDENTITY_PROVIDERS
- SVV_INTEGRATION
- SVV_INTEGRATION_TABLE_STATE
- SVV_INTERLEAVED_COLUMNS
- SVV_LANGUAGE_PRIVILEGES

- SVV_MASKING_POLICY
- SVV_ML_MODEL_INFO
- SVV_ML_MODEL_PRIVILEGES
- SVV_MV_DEPENDENCY
- SVV_MV_INFO
- SVV_QUERY_INFLIGHT
- SVV_QUERY_STATE
- SVV_REDSHIFT_COLUMNS
- SVV_REDSHIFT_DATABASE
- SVV_REDSHIFT_FUNCTIONS
- SVV_REDSHIFT_SCHEMA_QUOTA
- SVV_REDSHIFT_SCHEMAS
- SVV_REDSHIFT_TABLES
- SVV_RELATION_PRIVILEGES
- SVV_RLS_APPLIED_POLICY
- SVV_RLS_ATTACHED_POLICY
- SVV_RLS_POLICY
- SVV_RLS_RELATION
- SVV_ROLE_GRANTS
- SVV_ROLES
- SVV_SCHEMA_PRIVILEGES
- SVV_SCHEMA_QUOTA_STATE
- SVV_SYSTEM_PRIVILEGES
- SVV_TABLE_INFO
- SVV_TABLES
- SVV_TRANSACTIONS
- SVV_USER_GRANTS
- SVV_USER_INFO
- SVV_VACUUM_PROGRESS

- SVV_VACUUM_SUMMARY

SVV_ACTIVE_CURSORS

SVV_ACTIVE_CURSORS visualizza i dettagli dei cursori correntemente aperti. Per ulteriori informazioni, consulta DECLARE.

SVV_ACTIVE_CURSORS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema. Un utente può visualizzare soltanto i cursori che ha aperto. un utente con privilegi avanzati può visualizzare tutti i cursori.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'ID dell'utente che ha creato il cursore.
cursor_name	varchar(128)	Nome del cursore.
transaction_id	bigint(128)	L'ID della transazione.
session_id	integer	L'ID del processo con il cursore attivo.
declare_time	timestamp	Ora in cui il cursore è stato dichiarato.
total_bytes	bigint	Dimensione in byte nel set di risultati del cursore.
total_rows	bigint	Numero di righe nel set di risultati del cursore.
fetched_rows	bigint	Numero di righe correntem ente recuperate dal set di risultati del cursore.

Nome colonna	Tipo di dati	Descrizione
cursor_storage_limit_used_percent	integer	Il parametro indica la percentuale di spazio su disco attualmente utilizzato dal cursore.

SVV_ALL_COLUMNS

Utilizza SVV_ALL_COLUMNS per visualizzare un'unione di colonne delle tabelle Amazon Redshift, come mostrato nelle tabelle SVV_REDSHIFT_COLUMNS e l'elenco consolidato di tutte le colonne esterne di tutte le tabelle esterne. Per informazioni sulle colonne Amazon Redshift, consultare SVV_REDSHIFT_COLUMNS.

SVV_ALL_COLUMNS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	varchar(128)	Nome del database.
schema_name	varchar(128)	Il nome dello schema.
table_name	varchar(128)	Nome della tabella.
column_name	varchar(128)	Il nome della colonna.
ordinal_position	integer	La posizione della colonna nella tabella.
column_default	varchar(4000)	Il valore predefinito della colonna.

Nome colonna	Tipo di dati	Descrizione
is_nullable	varchar(3)	Un valore che Indica se la colonna è nullable. I valori possibili sono sì e no.
data_type	varchar(128)	Il tipo di dati della colonna.
character_maximum_length	integer	Il numero massimo di caratteri nella colonna.
numeric_precision	integer	La precisione numerica.
numeric_scale	integer	La scala numerica.
remarks	varchar(256)	Osservazioni.

Query di esempio

L'esempio seguente restituisce l'output di SVV_ALL_COLUMNS.

```
SELECT *
FROM svv_all_columns
WHERE database_name = 'tickit_db'
    AND TABLE_NAME = 'tickit_sales_redshift'
ORDER BY COLUMN_NAME,
    SCHEMA_NAME
LIMIT 5;

 database_name | schema_name |     table_name        | column_name | ordinal_position
 | column_default | is_nullable | data_type | character_maximum_length |
 numeric_precision | numeric_scale | remarks
 --------------+-------------+-----------------------+-------------+-----------------
+----------------+-------------+-----------+--------------------------
+------------------+--------------+---------
   tickit_db   |    public   | tickit_sales_redshift |    buyerid  |        4         |
               |      NO     |  integer  |                          |         32
  |       0       |
   tickit_db   |    public   | tickit_sales_redshift | commission  |        9         |
               |     YES     |  numeric  |                          |          8
  | 2       |
```

```
 tickit_db   |    public   | tickit_sales_redshift |    dateid   |        7         |
              |      NO     |  smallint |                          |         16
|       0       |
 tickit_db   |    public   | tickit_sales_redshift |   eventid   |        5         |
              |      NO     |  integer  |                          |         32
|       0       |
 tickit_db   |    public   | tickit_sales_redshift |    listid   |        2         |
              |      NO     |  integer  |                          |         32
|       0       |
```

SVV_ALL_SCHEMAS

Utilizza SVV_ALL_SCHEMAS per visualizzare un'unione di schemi Amazon Redshift come mostrato in SVV_REDSHIFT_SCHEMAS e l'elenco consolidato di tutti gli schemi esterni di tutti i database. Per ulteriori informazioni sugli schemi Amazon Redshift, consultare SVV_REDSHIFT_SCHEMAS.

SVV_ALL_SCHEMAS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	varchar(128)	Il nome del database in cui è presente lo schema.
schema_name	varchar(128)	Il nome dello schema.
schema_owner	integer	L'ID utente del proprietario dello schema. Per informazi oni sugli ID utente, consulta PG_USER_INFO.
schema_type	varchar(128)	Il tipo di schema. I valori possibili sono schemi esterni, locali e condivisi.
schema_acl	varchar(128)	La stringa che definisce le autorizzazioni per l'utente o il

Nome colonna	Tipo di dati	Descrizione
		gruppo di utenti specificato per lo schema.
source_database	varchar(128)	Il nome del database di origine per lo schema esterno.
schema_option	varchar(256)	Le opzioni dello schema. Questo è un attributo di schema esterno.

Query di esempio

L'esempio seguente restituisce l'output di SVV_ALL_SCHEMAS.

```
SELECT *
FROM svv_all_schemas
WHERE database_name = 'tickit_db'
ORDER BY database_name,
    SCHEMA_NAME;


 database_name |    schema_name     | schema_owner | schema_type | schema_acl |
 source_database | schema_option
---------------+--------------------+--------------+-------------+------------
+-----------------+---------------
   tickit_db   |       public       |       1      |   shared    |            |
         |
```

SVV_ALL_TABLES

Utilizza SVV_ALL_TABLES per visualizzare un'unione di tabelle Amazon Redshift, come mostrato nelle tabelle SVV_ALL_TABLES e l'elenco consolidato di tutte le tabelle esterne di tutti gli schemi esterni. Per informazioni sulle tabelle Amazon Redshift, consultare SVV_REDSHIFT_TABLES.

SVV_ALL_TABLES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	varchar(128)	Il nome del database in cui è presente la tabella.
schema_name	varchar(128)	Il nome dello schema per la tabella.
table_name	varchar(128)	Nome della tabella.
table_acl	varchar(128)	La stringa che definisce l'autorizzazione per l'utente o il gruppo di utenti specificato per la tabella.
table_type	varchar(128)	Il tipo di tabella. I valori possibili sono viste, tabelle di base, tabelle esterne e tabelle condivise.
remarks	varchar(256)	Osservazioni.

Query di esempio

L'esempio seguente restituisce l'output di SVV_ALL_TABLES.

```
SELECT *
FROM svv_all_tables
WHERE database_name = 'tickit_db'
ORDER BY TABLE_NAME,
    SCHEMA_NAME
LIMIT 5;

 database_name | schema_name |        table_name        | table_type | table_acl |
 remarks
---------------+-------------+--------------------------+------------+-----------
+---------
   tickit_db   |    public   | tickit_category_redshift |    TABLE   |           |
```

```
 tickit_db   |    public   |   tickit_date_redshift   |    TABLE   |           |
 tickit_db   |    public   |   tickit_event_redshift  |    TABLE   |           |
 tickit_db   |    public   | tickit_listing_redshift  |    TABLE   |           |
 tickit_db   |    public   |   tickit_sales_redshift  |    TABLE   |           |
```

Se il valore table_acl è nullo, non sono stati concessi esplicitamente privilegi di accesso alla tabella corrispondente.

SVV_ALTER_TABLE_RECOMMENDATIONS

Registra i suggerimenti correnti di Amazon Redshift Advisor per le tabelle. Questa visualizzazione mostra i suggerimenti per tutte le tabelle, indipendentemente dal fatto che siano definite o meno per l'ottimizzazione automatica. Per verificare se una tabella è definita per l'ottimizzazione automatica, consultare SVV_TABLE_INFO. Le voci sono visualizzate solo per le tabelle visibili nel database della sessione corrente. Dopo che è stato applicato un suggerimento (da Amazon Redshift o da te), non viene più visualizzato nella visualizzazione.

SVV_ALTER_TABLE_RECOMMENDATIONS è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
tipo	character (30)	Il tipo di suggerimento. I valori possibili sono distkey e sortkey.
database	character (128)	Nome del database.
table_id	integer	L'identificatore della tabella.
group_id	integer	Il numero di gruppo di una serie di suggerimenti. Per avere il massimo beneficio, dovrebbero essere applicati tutti i suggerimenti di un gruppo. I valori possibili sono -1 per un suggerimento chiave di ordinamento e un numero maggiore di zero per un suggerimento chiave di distribuzione.

Nome colonna	Tipo di dati	Descrizione
ddl	character (1024)	L'istruzione SQL che deve essere eseguita per applicare il suggerimento.
auto_eligible	character (1)	Il valore indica se il suggerimento è idoneo per l'esecuzione automatica di Amazon Redshift. Se questo valore è `t` allora l'indicazione è true, se è `f` allora è false.

Query di esempio

Nell'esempio seguente, le righe nel risultato mostrano i suggerimenti per la chiave di distribuzione e la chiave di ordinamento. Le righe mostrano anche se i suggerimenti sono idonei per Amazon Redshift per applicarli automaticamente.

```
select type, database, table_id, group_id, ddl, auto_eligible
from svv_alter_table_recommendations;
```

```
 type      | database | table_id | group_id | ddl

                   | auto_eligible
diststyle | db0      | 117884   | 2        | ALTER TABLE "sch"."dp21235_tbl_1" ALTER
DISTSTYLE KEY DISTKEY "c0"
                     | f
diststyle | db0      | 117892   | 2        | ALTER TABLE "sch"."dp21235_tbl_1" ALTER
DISTSTYLE KEY DISTKEY "c0"
                     | f
diststyle | db0      | 117885   | 1        | ALTER TABLE "sch"."catalog_returns"
ALTER DISTSTYLE KEY DISTKEY "cr_sold_date_sk", ALTER COMPOUND SORTKEY
("cr_sold_date_sk","cr_returned_time_sk") | t
sortkey   | db0      | 117890   | -1       | ALTER TABLE "sch"."customer_addresses"
ALTER COMPOUND SORTKEY ("ca_address_sk")
                      | t
```

SVV_ATTACHED_MASKING_POLICY

Utilizza SVV_ATTACHED_MASKING_POLICY per visualizzare tutte le relazioni e i ruoli/gli utenti che hanno policy collegate al database attualmente connesso.

Solo gli utenti con privilegi avanzati e gli utenti con il ruolo `sys:secadmin` possono visualizzare SVV_ATTACHED_MASKING_POLICY. Gli utenti con privilegi normali vedranno 0 righe.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
nome_policy	text	Nome della policy di mascheramento collegata alla tabella.
schema_name	text	Schema della tabella a cui è collegata la policy.
table_name	text	Nome della tabella a cui è collegata la policy.
table_type	text	Tipo di tabella a cui è collegata la policy.
grantor	text	Nome dell'utente che ha collegato la policy.
grantee	text	Nome dell'utente/del ruolo a cui è collegata la policy.
grantee_type	text	Il tipo di assegnatario. Può trattarsi di role (ruolo), user (utente) o public (pubblico).
priority	int	Priorità della policy collegata.
input_columns	text	Attributi della colonna di input della policy collegata.

Nome colonna	Tipo di dati	Descrizione
output_columns	text	Attributi della colonna di output della policy collegata.

Funzioni interne

SVV_ATTACHED_MASKING_POLICY supporta le seguenti funzioni interne:

mask_get_policy_per_ruolo_on_column

Ottieni la policy con la massima priorità applicabile a una determinata coppia ruolo/colonna.

Sintassi

```
mask_get_policy_for_role_on_column
                        (relschema,
                        relname,
                        colname,
                        rolename);
```

Parametri

relschema

Nome dello schema in cui si trova la policy.

relname

Il nome della tabella in cui si trova la policy.

colname

Il nome della colonna a cui è collegata la policy.

rolename

Nome del ruolo a cui è collegata la policy.

mask_get_policy_per_user_on_column

Ottieni la policy con la massima priorità che si applica a una determinata coppia colonna/utente.

Sintassi

```
mask_get_policy_for_user_on_column
                        (relschema,
                        relname,
                        colname,
                        username);
```

Parametri

relschema

Nome dello schema in cui si trova la policy.

relname

Il nome della tabella in cui si trova la policy.

colname

Il nome della colonna a cui è collegata la policy.

rolename

Nome dell'utente che ha collegata la policy.

SVV_COLUMNS

Utilizzare SVV_COLUMNS per visualizzare le informazioni di catalogo sulle colonne delle tabelle e delle visualizzazioni locali ed esterne, comprese le visualizzazioni con associazione tardiva.

SVV_COLUMNS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

La visualizzazione SVV_COLUMNS si unisce ai metadati di tabella dalle Tabelle di catalogo di sistema (tabelle con un prefisso PG) e la visualizzazione di sistema SVV_EXTERNAL_COLUMNS. Le tabelle del catalogo di sistema descrivono le tabelle del database Amazon Redshift. SVV_EXTERNAL_COLUMNS descrive le tabelle esterne utilizzate con Amazon Redshift Spectrum.

Tutti gli utenti possono vedere tutte le righe delle tabelle del catalogo di sistema. Gli utenti normali possono vedere le definizioni delle colonne dalla visualizzazione SVV_EXTERNAL_COLUMNS solo per le tabelle esterne a cui è stato consentito l'accesso. Sebbene gli utenti normali possano vedere

i metadati delle tabelle nelle tabelle del catalogo di sistema, possono selezionare i dati delle tabelle definite dall'utente solo se sono i proprietari della tabella o se è stato loro consentito l'accesso.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
table_catalog	text	Il nome del catalogo in cui si trova la tabella.
table_schema	text	Il nome dello schema per la tabella.
table_name	text	Nome della tabella.
column_name	text	Il nome della colonna.
ordinal_position	int	La posizione della colonna nella tabella.
column_default	text	Il valore predefinito della colonna.
is_nullable	text	Un valore che Indica se la colonna è nullable.
data_type	text	Il tipo di dati della colonna.
character_maximum_length	int	Il numero massimo di caratteri nella colonna.
numeric_precision	int	La precisione numerica. Se la colonna data_type è numerica, essa restituisce il numero di cifre significative dell'intero valore.
numeric_precision_radix	int	La radice di precisione numerica. Se la colonna data_type è numerica, essa

Nome colonna	Tipo di dati	Descrizione
		restituisce la base delle colonne numeric_precision e numeric_scale.
numeric_scale	int	La scala numerica. Se la colonna data_type è numerica, essa restituisce il numero di cifre significative nel valore decimale.
datetime_precision	int	La precisione datetime.
interval_type	text	Il tipo di intervallo.
interval_precision	text	La precisione di intervallo.
character_set_catalog	text	Il catalogo di impostazione carattere.
character_set_schema	text	Lo schema di impostazione carattere.
character_set_name	text	Il nome di impostazione carattere.
collation_catalog	text	Il catalogo di confronto.
collation_schema	text	Lo schema di confronto.
collation_name	text	Il nome di confronto.
domain_name	text	Il nome di dominio.
remarks	text	Osservazioni.

SVV_COLUMN_PRIVILEGES

Utilizza SVV_COLUMN_PRIVILEGES per visualizzare le autorizzazioni di colonna concesse esplicitamente a utenti, ruoli e gruppi nel database corrente.

SVV_COLUMN_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
namespace_name	text	Il nome dello spazio dei nomi in cui è presente la relazione specificata.
relation_name	text	Il nome della relazione.
column_name	text	Il nome della colonna.
privilege_type	text	Il tipo di autorizzazione. I valori possibili sono SELECT o UPDATE.
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente, ruolo, gruppo o pubblico.

Query di esempio

L'esempio seguente mostra il risultato di SVV_COLUMN_PRIVILEGES.

```
SELECT
 namespace_name,relation_name,COLUMN_NAME,privilege_type,identity_name,identity_type
FROM svv_column_privileges WHERE relation_name = 'lineitem';

 namespace_name | relation_name | column_name | privilege_type | identity_name |
 identity_type
----------------+---------------+-------------+----------------+---------------
+----------------
    public      |   lineitem    | l_orderkey  |     SELECT     |    reguser    |
 user
    public      |   lineitem    | l_orderkey  |     SELECT     |     role1     |
 role
    public      |   lineitem    | l_partkey   |     SELECT     |    reguser    |
 user
    public      |   lineitem    | l_partkey   |     SELECT     |     role1     |
 role
```

SVV_DATABASE_PRIVILEGES

Utilizza SVV_DATABASE_PRIVILEGES per visualizzare le autorizzazioni del database concesse esplicitamente a utenti, ruoli e gruppi nel cluster Amazon Redshift.

SVV_DATABASE_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	text	Nome del database.
privilege_type	text	Il tipo di autorizzazione. I valori possibili sono USAGE, CREATE o TEMP.

Nome colonna	Tipo di dati	Descrizione
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente, ruolo, gruppo o pubblico.
admin_option	booleano	Un valore che indica se l'utente può concedere l'autorizzazione ad altri utenti e ruoli. È sempre false per il ruolo e il tipo di identità del gruppo.

Query di esempio

L'esempio seguente mostra il risultato di SVV_DATABASE_PRIVILEGES.

```
SELECT database_name,privilege_type,identity_name,identity_type,admin_option FROM
 svv_database_privileges
WHERE database_name = 'test_db';

 database_name | privilege_type | identity_name | identity_type | admin_option
---------------+----------------+---------------+---------------+--------------
     test_db   |     CREATE     |     reguser   |      user     |     False
     test_db   |     CREATE     |      role1    |      role     |     False
     test_db   |     TEMP       |      public   |      public   |     False
     test_db   |     TEMP       |      role1    |      role     |     False
```

SVV_DATASHARE_PRIVILEGES

Utilizza SVV_DATASHARE_PRIVILEGES per visualizzare le autorizzazioni dell'unità di condivisione dati esplicitamente concesse a utenti, ruoli e gruppi nel cluster Amazon Redshift.

SVV_DATASHARE_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
datashare_name	text	Il nome dell'unità di condivisione dati.
privilege_type	text	Il tipo di autorizzazione. I valori possibili sono ALTER o SHARE.
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente, ruolo, gruppo o pubblico.
admin_option	booleano	Un valore che indica se l'utente può concedere l'autorizzazione ad altri utenti e ruoli. È sempre false per il ruolo e il tipo di identità del gruppo.

Query di esempio

L'esempio seguente mostra il risultato di SVV_DATASHARE_PRIVILEGES.

```
SELECT datashare_name,privilege_type,identity_name,identity_type,admin_option FROM
 svv_datashare_privileges
WHERE datashare_name = 'demo_share';

 datashare_name | privilege_type |  identity_name | identity_type | admin_option
----------------+----------------+----------------+---------------+--------------
   demo_share   |     ALTER      |    superuser   |     user      |   False
   demo_share   |     ALTER      |    reguser     |     user      |   False
```

SVV_DATASHARES

Utilizza SVV_DATASHARES per visualizzare un elenco di unità di condivisione dati creati sul cluster e unità di condivisione dati condivise con il cluster.

SVV_DATASHARES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Proprietari dell'unità di condivisione dati
- Utenti con autorizzazioni ALTER o USAGE per un'unità di condivisione dati

Gli altri utenti non possono visualizzare alcuna riga. Per informazioni sulle autorizzazioni ALTER e USAGE, consulta GRANT.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
share_name	varchar(128)	Il nome di una unità di condivisione dati.
share_id	integer	L'ID dell'unità di condivisione dati.
share_owner	integer	Il proprietario dell'unità di condivisione dati.
source_database	varchar(128)	Il database di origine per questa unità di condivisione dati.
consumer_database	varchar(128)	Il database del consumer creato da questa unità di condivisione dati.
share_type	varchar(8)	Il tipo di unità di condivisione dati. I valori possibili sono INBOUND e OUTBOUND.

Nome colonna	Tipo di dati	Descrizione
createdate	timestamp without time zone	La data in cui è stata creata l'unità di condivisione dati.
is_publicaccessible	booleano	La proprietà che specifica se una unità di condivisione dati può essere condivisa in un cluster accessibile pubblicam ente.
share_acl	varchar(256)	La stringa che definisce le autorizzazioni per l'utente o il gruppo di utenti specificato per l'unità di condivisione dati.
producer_account	varchar(16)	L'ID dell'account del produttor e di unità di condivisione dati.
producer_namespace	varchar(64)	L'identificatore del cluster univoco per il cluster di produttori dell'unità di condivisi one dati.
managed_by	varchar(64)	La proprietà che specifica il AWS servizio che gestisce il datashare.

Note per l'utilizzo

Recupero di metadati aggiuntivi: utilizzando il numero intero restituito nella `share_owner` colonna, puoi unirti a in per ottenere dati sul proprietario del `usesysid` SVL_USER_INFO datashare. Ciò include il nome e le proprietà aggiuntive.

Query di esempio

L'esempio seguente restituisce l'output di SVV_DATASHARES.

```
SELECT share_owner, source_database, share_type, is_publicaccessible
```

```
FROM svv_datashares
WHERE share_name LIKE 'tickit_datashare%'
AND source_database = 'dev';

  share_owner | source_database | share_type  | is_publicaccessible
--------------+-----------------+-------------+----------------------
     100      |      dev        |   OUTBOUND  |        True
(1 rows)
```

L'esempio seguente restituisce l'output di SVV_DATASHARES per la condivisione di dati in uscita.

```
SELECT share_name, share_owner, btrim(source_database), btrim(consumer_database),
 share_type, is_publicaccessible, share_acl, btrim(producer_account),
 btrim(producer_namespace), btrim(managed_by) FROM svv_datashares WHERE share_type =
 'OUTBOUND';

   share_name   | share_owner | source_database | consumer_database | share_type |
 is_publicaccessible | share_acl | producer_account|         producer_namespace
   | managed_by
----------------+-------------+-----------------+-------------------
+------------+---------------------+-----------+-----------------
+--------------------------------------+------------
    salesshare  |      1      |       dev       |                   |  OUTBOUND  |
   True         |           |   123456789012  | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d |

 marketingshare |      1      |       dev       |                   |  OUTBOUND  |
   True         |           |   123456789012  | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d |
```

L'esempio seguente restituisce l'output di SVV_DATASHARES per la condivisione di dati in entrata.

```
SELECT share_name, share_owner, btrim(source_database), btrim(consumer_database),
 share_type, is_publicaccessible, share_acl, btrim(producer_account),
 btrim(producer_namespace), btrim(managed_by) FROM svv_datashares WHERE share_type =
 'INBOUND';

  share_name    | share_owner | source_database | consumer_database | share_type |
 is_publicaccessible | share_acl | producer_account |         producer_namespace
    | managed_by
----------------+-------------+-----------------+-------------------
+------------+---------------------+-----------+-----------------
+--------------------------------------+------------
```

```
 salesshare    |             |                  |                    |  INBOUND   |
  False         |           |  123456789012    | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d
|
marketingshare |             |                  |                    |  INBOUND   |
 False         |           |  123456789012    | 13b8833d-17c6-4f16-8fe4-1a018f5ed00d |
ADX
```

SVV_DATASHARE_CONSUMERS

Utilizza SVV_DATASHARE_CONSUMERS per visualizzare un elenco di consumer per una unità di condivisione dati creata in un cluster.

SVV_DATASHARE_CONSUMERS è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Proprietari dell'unità di condivisione dati
- Utenti con autorizzazioni ALTER o USAGE per un'unità di condivisione dati

Gli altri utenti non possono visualizzare alcuna riga. Per informazioni sulle autorizzazioni ALTER e USAGE, consulta GRANT.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
share_name	varchar(128)	Il nome dell'unità di condivisione dati.
consumer_account	varchar(16)	L'ID account per il consumer dell'unità di condivisione dati.
consumer_namespace	varchar(64)	L'identificatore del cluster univoco per il cluster di consumer dell'unità di condivisione dati.
share_date	timestamp without time zone	La data di condivisione dell'unità di condivisione dati.

Query di esempio

L'esempio seguente restituisce l'output di SVV_DATASHARE_CONSUMERS.

```
SELECT count(*)
FROM svv_datashare_consumers
WHERE share_name LIKE 'tickit_datashare%';

1
```

SVV_DATASHARE_OBJECTS

Utilizza SVV_DATASHARE_OBJECTS per visualizzare un elenco di oggetti in tutte le unità di condivisione dati create sul cluster o condivise con il cluster.

SVV_DATASHARE_OBJECTS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni sulla visualizzazione di un elenco di unità di condivisione dati, consulta SVV_DATASHARES.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
share_type	varchar(8)	Il tipo di unità di condivisi one dati specificato. I valori possibili sono OUTBOUND e INBOUND.
share_name	varchar(128)	Il nome dell'unità di condivisi one dati.
object_type	varchar(64)	Il tipo di oggetto specificato. I valori possibili sono schemi, tabelle, viste, viste di associazi one tardiva, viste materiali zzate e funzioni.

Nome colonna	Tipo di dati	Descrizione
object_name	varchar(512)	Il nome dell'oggetto. Il nome dell'oggetto si estende per includere il nome dello schema, ad esempio schema1.t1.
producer_account	varchar(16)	L'ID dell'account del produttore di unità di condivisione dati.
producer_namespace	varchar(64)	L'identificatore del cluster univoco per il cluster di produttori dell'unità di condivisione dati.
include_new	booleano	La proprietà che specifica se aggiungere all'unità di condivisione dati tabelle, viste o funzioni definite dall'utente (FDU) SQL future create nell'unità di condivisione dati specificata. Questo parametro è rilevante solo per le unità di condivisione dati OUTBOUND e solo per i tipi di schema nell'unità.

Query di esempio

L'esempio seguente restituisce l'output di SVV_DATASHARE_OBJECTS.

```
SELECT share_type,
    btrim(share_name)::varchar(16) AS share_name,
    object_type,
    object_name
FROM svv_datashare_objects
WHERE share_name LIKE 'tickit_datashare%'
```

```
AND object_name LIKE '%tickit%'
ORDER BY object_name
LIMIT 5;

 share_type |     share_name     | object_type |          object_name
------------+--------------------+-------------+--------------------------------
 OUTBOUND   |  tickit_datashare  |    table    |  public.tickit_category_redshift
 OUTBOUND   |  tickit_datashare  |    table    |  public.tickit_date_redshift
 OUTBOUND   |  tickit_datashare  |    table    |  public.tickit_event_redshift
 OUTBOUND   |  tickit_datashare  |    table    |  public.tickit_listing_redshift
 OUTBOUND   |  tickit_datashare  |    table    |  public.tickit_sales_redshift
```

```
SELECT * FROM SVV_DATASHARE_OBJECTS WHERE share_name like 'sales%';

share_type | share_name | object_type | object_name  | producer_account |
 producer_namespace          | include_new
-----------+------------+-------------+--------------+------------------
+--------------------------------------+-------------
 OUTBOUND  | salesshare | schema      | public       | 123456789012     |
 13b8833d-17c6-4f16-8fe4-1a018f5ed00d |      t
 OUTBOUND  | salesshare | table       | public.sales | 123456789012     |
 13b8833d-17c6-4f16-8fe4-1a018f5ed00d |
```

SVV_DEFAULT_PRIVILEGES

Utilizzare SVV_DEFAULT_PRIVILEGES per visualizzare i privilegi predefiniti a cui l'utente ha accesso in un cluster Amazon Redshift.

SVV_DEFAULT_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le autorizzazioni predefinite assegnate.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
schema_name	text	Il nome dello schema.
object_type	text	Il tipo di oggetto. I valori possibili sono RELATION, FUNCTION o PROCEDURE.
owner_id	integer	L'ID del proprietario. Il valore possibile è l'ID utente.
owner_name	text	Il nome del proprietario.
owner_type	text	Il tipo di proprietario. Il valore possibile è l'utente.
privilege_type	text	Il tipo di privilegio. I valori possibili sono INSERT, SELECT, UPDATE, DELETE, RULE, REFERENCES TRIGGER, DROP ed EXECUTE.
grantee_id	integer	L'ID dell'assegnatario. I valori possibili sono ID utente, ID ruolo e ID gruppo.
grantee_type	text	Il tipo di assegnatario. I valori possibili sono utente, ruolo, gruppo e pubblico.
admin_option	booleano	Un valore che indica se l'utente può concedere le autorizzazioni ad altri utenti e ruoli. Il valore è sempre false per il ruolo e il tipo del gruppo.

Query di esempio

L'esempio seguente restituisce l'output di SVV_DEFAULT_PRIVILEGES.

```
SELECT * from svv_default_privileges;

 schema_name |    object_type    | owner_id | owner_name | owner_type | privilege_type
 | grantee_id | grantee_name | grantee_type | admin_option
-------------+-------------------+--------- +------------+------------+---------------
+------------+--------------+--------------+-------------+
```

```
  public    |     RELATION      |    106   |     u1     |    user    |     UPDATE
|     107    |      u2      |     user     |      f      |
  public    |     RELATION      |    106   |     u1     |    user    |     SELECT
|     107    |      u2      |     user     |      f      |
```

SVV_DISKUSAGE

Amazon Redshift crea la vista di sistema SVV_DISKUSAGE unendo le tabelle STV_TBL_PERM e STV_BLOCKLIST. La visualizzazione SVV_DISKUSAGE contiene informazioni sull'allocazione dati per le tabelle di un database.

Per determinare il numero di blocchi del disco allocati per database, tabella, sezione o colonna, utilizzare query di aggregazione con SVV_DISKUSAGE, come mostrano gli esempi seguenti. Ogni blocco di dati utilizza 1 MB. È possibile anche utilizzare STV_PARTITIONS per visualizzare informazioni di riepilogo sull'utilizzo del disco.

SVV_DISKUSAGE è visibile solo per gli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> Questa vista è disponibile solo per le query sui cluster con provisioning.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
db_id	integer	ID database.
nome	character (72)	Nome tabella.
sezione	integer	Sezione dati allocata alla tabella.
col	integer	Indice in base zero della colonna. A ogni tabella che crei vengono aggiunte tre colonne nascoste: INSERT_XID, DELETE_XID e ROW_ID

Nome colonna	Tipo di dati	Descrizione
		(OID). Una tabella con 3 colonne definite dall'utente contiene 6 colonne effettive e le colonne definite dall'utente sono numerate 0, 1 e 2. In questo esempio, le colonne INSERT_XID, DELETE_XID e ROW_ID sono numerate 3, 4 e 5 rispettivamente.
tbl	integer	ID tabella.
blocknum	integer	ID del blocco di dati.
num_values	integer	Numero di valori contenuti nel blocco.
minvalue	bigint	Valore minimo contenuto nel blocco.
maxvalue	bigint	Valore massimo contenuto nel blocco.
sb_pos	integer	Identificatore interno della posizione del super blocco sul disco.
pinned	integer	Indica se il blocco è aggiunto o meno nella memoria come parte del precaricamento. 0 = false; 1 = true. Il valore predefinito è false.
on_disk	integer	Indica se il blocco è archiviato automaticamente sul disco o meno. 0 = false; 1 = true. Il valore predefinito è false.
modified	integer	Indica se il blocco è stato modificato o meno. 0 = false; 1 = true. Il valore predefinito è false.
hdr_modified	integer	Indica se l'intestazione del blocco è stata modificata o meno. 0 = false; 1 = true. Il valore predefinito è false.
unsorted	integer	Indica se il blocco è non ordinato o meno. 0 = false; 1 = true. Il valore predefinito è true.
tombstone	integer	Per uso interno.
preferred_diskno	integer	Numero di disco su cui il blocco deve trovarsi, salvo in caso di errore del disco. Dopo la correzione del disco, il blocco ritorna su questo disco.

Nome colonna	Tipo di dati	Descrizione
temporary	integer	Indica se il blocco contiene o meno dati temporanei, ad esempio di una tabella temporanea o di risultati intermedi delle query. 0 = false; 1 = true. Il valore predefinito è false.
newblock	integer	Indica se un blocco è nuovo (true) o se non ne è mai stato eseguito il commit sul disco (false). 0 = false; 1 = true.

Query di esempio

SVV_DISKUSAGE contiene una riga per blocco del disco allocato, di conseguenza una query che seleziona tutte le righe potenzialmente restituisce un numero molto grande di righe. Si consiglia di utilizzare solo query di aggregazione con SVV_DISKUSAGE.

Restituire il maggior numero di blocchi mai allocati alla colonna 6 nella tabella USERS (la colonna EMAIL):

```
select db_id, trim(name) as tablename, max(blocknum)
from svv_diskusage
where name='users' and col=6
group by db_id, name;

db_id  | tablename | max
--------+-----------+-----
175857 | users     |   2
(1 row)
```

La seguente query restituisce risultati simili per tutte le colonne in una tabella grande a 10 colonne chiamata SALESNEW. (Le ultime tre righe, per le colonne da 10 a 12, si riferiscono alle colonne dei metadati nascosti).

```
select db_id, trim(name) as tablename, col, tbl, max(blocknum)
from svv_diskusage
where name='salesnew'
group by db_id, name, col, tbl
order by db_id, name, col, tbl;

db_id  | tablename  | col |  tbl   | max
```

```
--------+------------+-----+--------+-----
175857 | salesnew   |   0 | 187605 | 154
175857 | salesnew   |   1 | 187605 | 154
175857 | salesnew   |   2 | 187605 | 154
175857 | salesnew   |   3 | 187605 | 154
175857 | salesnew   |   4 | 187605 | 154
175857 | salesnew   |   5 | 187605 |  79
175857 | salesnew   |   6 | 187605 |  79
175857 | salesnew   |   7 | 187605 | 302
175857 | salesnew   |   8 | 187605 | 302
175857 | salesnew   |   9 | 187605 | 302
175857 | salesnew   |  10 | 187605 |   3
175857 | salesnew   |  11 | 187605 |   2
175857 | salesnew   |  12 | 187605 | 296
(13 rows)
```

SVV_EXTERNAL_COLUMNS

Utilizzare SVV_EXTERNAL_COLUMNS per visualizzare i dettagli delle colonne nelle tabelle esterne. Utilizza SVV_EXTERNAL_COLUMNS anche per le query tra database per visualizzare i dettagli su tutte le colonne della tabella nei database non connessi a cui gli utenti hanno accesso.

SVV_EXTERNAL_COLUMNS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
redshift_database_name	text	Nome del database Amazon Redshift.
schemaname	text	Il nome dello schema esterno di Amazon Redshift per la tabella esterna.
tablename	text	Il nome della tabella esterna.
columnname	text	Il nome della colonna.

Nome colonna	Tipo di dati	Descrizione
external_type	text	Il tipo di dati della colonna.
columnnum	integer	Il numero della colonna esterna, partendo da 1.
part_key	integer	Se la colonna è una chiave di partizione, l'ordine della chiave. Se la colonna non è una chiave di partizione, il valore è **0**.
is_nullable	text	Definisce se una colonna è nullable o meno. Alcuni valori sono `true`, `false` o " " stringa vuota che non rappresenta alcuna informazione.

SVV_EXTERNAL_DATABASES

Utilizzare SVV_EXTERNAL_DATABASES per visualizzare i dettagli dei database esterni.

SVV_EXTERNAL_DATABASES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
eskind	integer	Il tipo di catalogo esterno per il database; **1** indica un catalogo di dati, **2** indica un metastore Hive.

Nome colonna	Tipo di dati	Descrizione
esoptions	text	Dettagli del catalogo in cui risiede il database.
databasename	text	Il nome del database nel catalogo esterno.
posizione	text	La posizione del database.
parameters	text	Parametri database.

SVV_EXTERNAL_PARTITIONS

Utilizzare SVV_EXTERNAL_PARTITIONS per visualizzare i dettagli delle partizioni nelle tabelle esterne.

SVV_EXTERNAL_PARTITIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema..

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
schemaname	text	Il nome dello schema esterno di Amazon Redshift per la tabella esterna con le partizioni specificate.
tablename	text	Il nome della tabella esterna.
values	text	Valori per la partizione.
posizione	text	La posizione della partizione. La colonna può avere una dimension e massima di 128 caratteri. I valori più lunghi vengono troncati.
input_format	text	Il formato dell'input.
output_format	text	Il formato dell'output.

Nome colonna	Tipo di dati	Descrizione
serialization_lib	text	La libreria di serializzazione.
serde_parameters	text	SerDe parametri.
compresso	integer	Un valore che indica se la partizione è compressa; **1** indica compressa, **0** indica non compressa.
parameters	text	Proprietà della partizione.

SVV_EXTERNAL_SCHEMAS

Utilizzare SVV_EXTERNAL_SCHEMAS per visualizzare le informazioni sugli schemi esterni. Per ulteriori informazioni, consulta CREATE EXTERNAL SCHEMA.

SVV_EXTERNAL_SCHEMAS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
esoid	oid	ID dello schema esterno.
eskind	smallint	Il tipo di catalogo esterno per lo schema esterno: 1 indica un catalogo di dati, 2 indica un metastore Hive, 3 indica una query federata per Aurora PostgreSQL o Amazon RDS PostgreSQL, 4 indica uno schema per un database Amazon Redshift locale, 5 indica uno schema per un database Amazon Redshift remoto, 6 indica per una tabella di sistema, 8 indica uno schema per database MySQL remoti, 9 indica uno schema per un flusso di dati Amazon

Nome colonna	Tipo di dati	Descrizione
		Kinesis e 10 indica un flusso di dati di streaming gestito da Amazon per Apache Kafka (Amazon MSK).
schemaname	nome	Nome schema esterno.
esowner	integer	ID utente del proprietario dello schema esterno.
databasename	text	Nome del database esterno.
esoptions	text	Opzioni dello schema esterno.

Esempio

Nell'esempio seguente vengono visualizzati i dettagli per gli schemi esterni.

```
select * from svv_external_schemas;

esoid  | eskind | schemaname | esowner | databasename | esoptions

-------+--------+------------+---------+--------------
+-------------------------------------------------------------
100133 |      1 | spectrum   |     100 | redshift     |
 {"IAM_ROLE":"arn:aws:iam::123456789012:role/mySpectrumRole"}
```

SVV_EXTERNAL_TABLES

Utilizzare SVV_EXTERNAL_TABLES per visualizzare i dettagli delle tabelle esterne. Per ulteriori informazioni, consultare CREATE EXTERNAL SCHEMA. Utilizza SVV_EXTERNAL_TABLES anche per le query tra database per visualizzare i metadati su tutte le tabelle nei database non connessi a cui gli utenti hanno accesso.

SVV_EXTERNAL_TABLES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
redshift_database_name	text	Nome del database Amazon Redshift.
schemaname	text	Il nome dello schema esterno di Amazon Redshift per la tabella esterna.
tablename	text	Il nome della tabella esterna.
tabletype	text	Il tipo di tabella. Alcuni valori sono TABLE, VIEW, MATERIALIZED VIEW o " " stringa vuota che non rappresenta alcuna informazi one.
posizione	text	La posizione della tabella.
input_format	text	Il formato dell'input
output_format	text	Il formato dell'output.
serialization_lib	text	La libreria di serializzazione.
serde_parameters	text	SerDe parametri.
compresso	integer	Un valore che indica se la tabella è compressa; **1** indica compressa, **0** indica non compressa.
parameters	text	Proprietà tabella.

Esempio

L'esempio seguente mostra i dettagli di svv_external_tables con un predicato sullo schema esterno utilizzato da una query federata.

```
select schemaname, tablename from svv_external_tables where schemaname = 'apg_tpch';
schemaname  | tablename
------------+-----------
apg_tpch    | customer
apg_tpch    | lineitem
apg_tpch    | nation
apg_tpch    | orders
apg_tpch    | part
apg_tpch    | partsupp
apg_tpch    | region
apg_tpch    | supplier
(8 rows)
```

SVV_FUNCTION_PRIVILEGES

Utilizza SVV_FUNCTION_PRIVILEGES per visualizzare le autorizzazioni di funzione concesse esplicitamente a utenti, ruoli e gruppi nel database corrente.

SVV_FUNCTION_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
namespace_name	text	Il nome dello spazio dei nomi in cui è presente la funzione specificata.
function_name	text	Il nome della funzione.

Nome colonna	Tipo di dati	Descrizione
argument_types	text	La stringa che rappresenta il tipo di argomento di input di una funzione.
privilege_type	text	Il tipo di autorizzazione. Il valore possibile è EXECUTE.
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente, ruolo, gruppo o pubblico.
admin_option	booleano	Un valore che indica se l'utente può concedere l'autorizzazione ad altri utenti e ruoli. È sempre false per il ruolo e il tipo di identità del gruppo.

Query di esempio

L'esempio seguente mostra il risultato di SVV_FUNCTION_PRIVILEGES.

```
SELECT
 namespace_name,function_name,argument_types,privilege_type,identity_name,identity_type,admin_o
 FROM svv_function_privileges
WHERE identity_name IN ('role1', 'reguser');

 namespace_name | function_name |       argument_types       | privilege_type |
 identity_name | identity_type | admin_option
----------------+---------------+----------------------------+----------------
+----------------+---------------+-------------
    public      | test_func1    | integer                    |    EXECUTE     |
 role1     |     role      |  False
    public      | test_func2    | integer, character varying |    EXECUTE     |
 reguser    |     user      |  False
```

SVV_GEOGRAPHY_COLUMNS

Utilizza SVV_GEOGRAPHY_COLUMNS per visualizzare l'elenco delle colonne GEOGRAPHY nel data warehouse. Questo elenco di colonne include colonne provenienti dalle unità di condivisione dati.

SVV_GEOGRAPHY_COLUMNS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
f_table_catalog	varchar(128)	Il nome del database in cui è presente la tabella contenente la colonna GEOGRAPHY.
f_table_schema	varchar(128)	Il nome dello schema in cui è presente la tabella contenente la colonna GEOGRAPHY.
f_table_name	varchar(128)	Il nome della tabella cui è presente la colonna GEOGRAPHY.
f_geography_column	varchar(128)	Il nome della colonna GEOGRAPHY.
coord_dimension	integer	Il numero di dimensioni dei dati GEOGRAPHY.
srid	integer	L'identificatore spaziale del sistema di riferimento (SRID) dei dati GEOGRAPHY.
tipo	varchar(128)	Il nome del tipo di dati GEOGRAPHY spaziali.

Query di esempio

L'esempio seguente mostra il risultato di SVV_GEOGRAPHY_COLUMNS.

```
SELECT * FROM svv_geography_columns;

f_table_catalog  | f_table_schema  | f_table_name  | f_geography_column  |
 coord_dimension | srid |  type
-----------------+-----------------+--------------+--------------------
+-----------------+------+-------------
dev              | public          | spatial_test  | test_geography      | 2
    | 0    | GEOGRAPHY
```

SVV_GEOMETRY_COLUMNS

Usa SVV_GEOMETRY_COLUMNS per visualizzare l'elenco delle colonne GEOMETRY nel data warehouse. Questo elenco di colonne include colonne provenienti dalle unità di condivisione dati.

SVV_GEOMETRY_COLUMNS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
f_table_catalog	varchar(128)	Il nome del database in cui è presente la tabella contenente la colonna GEOMETRY.
f_table_schema	varchar(128)	Il nome dello schema in cui è presente la tabella contenente la colonna GEOMETRY.
f_table_name	varchar(128)	Il nome della tabella cui è presente la colonna GEOMETRY.
f_geometry_column	varchar(128)	Il nome della colonna GEOMETRY.
coord_dimension	integer	Il numero di dimensioni dei dati GEOMETRY.
srid	integer	L'identificatore spaziale del sistema di riferimento (SRID) della colonna GEOMETRY.

Nome colonna	Tipo di dati	Descrizione
tipo	varchar(128)	Il nome del tipo GEOMETRY spaziale.

Query di esempio

L'esempio seguente mostra il risultato di SVV_GEOMETRY_COLUMNS.

```
SELECT * FROM svv_geometry_columns;

f_table_catalog  | f_table_schema  | f_table_name  | f_geometry_column   |
 coord_dimension | srid |  type
-----------------+-----------------+---------------+---------------------
+-----------------+------+--------------
dev              | public          | accomodations | shape               | 2
    | 0    | GEOMETRY
dev              | public          | zipcode       | wkb_geometry        | 2
    | 0    | GEOMETRY
```

SVV_IAM_PRIVILEGES

Utilizza SVV_IAM_PRIVILEGES per visualizzare i privilegi IAM concessi esplicitamente a utenti, ruoli e gruppi.

SVV_IAM_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le voci a cui hanno accesso.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
iam_arn	text	Nome dello spazio dei nomi.
command_type	text	Tipi di privilegi. I valori possibili sono COPY, UNLOAD, CREATE MODEL o EXTERNAL FUNCTION.
identity_id	integer	ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Nome dell'identità.
identity_type	text	Tipo di identità. I valori possibili sono utente, ruolo, gruppo o pubblico.

Query di esempio

L'esempio seguente mostra i risultati di SVV_IAM_PRIVILEGES.

```
SELECT * from SVV_IAM_PRIVILEGES ORDER BY IDENTITY_ID;
       iam_arn        | command_type | identity_id | identity_name | identity_type
----------------------+--------------+-------------+---------------+---------------
 default-aws-iam-role | COPY         |           0 | public        | public
 default-aws-iam-role | UNLOAD       |           0 | public        | public
 default-aws-iam-role | CREATE MODEL |           0 | public        | public
 default-aws-iam-role | EXFUNC       |           0 | public        | public
 default-aws-iam-role | COPY         |         106 | u1            | user
 default-aws-iam-role | UNLOAD       |         106 | u1            | user
 default-aws-iam-role | CREATE MODEL |         106 | u1            | user
 default-aws-iam-role | EXFUNC       |         106 | u1            | user
 default-aws-iam-role | COPY         |      118413 | r1            | role
 default-aws-iam-role | UNLOAD       |      118413 | r1            | role
 default-aws-iam-role | CREATE MODEL |      118413 | r1            | role
 default-aws-iam-role | EXFUNC       |      118413 | r1            | role
(12 rows)
```

SVV_IDENTITY_PROVIDERS

La vista SVV_IDENTITY_PROVIDERS restituisce il nome e le proprietà aggiuntive per i provider di identità. Per ulteriori informazioni sulla creazione di un provider di identità, consulta CREATE IDENTITY PROVIDER.

SVV_IDENTITY_PROVIDERS è visibile solo per gli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
uid	integer	l'ID univoco del provider di identità registrato.
nome	text	Il nome del provider di identità.
tipo	text	Il tipo di provider di identità.
instanceid	text	Il differenziatore univoco tra istanze dello stesso tipo.
namespc	text	Il nome dello spazio dei nomi del provider di identità.
params	text	L'oggetto JSON con i parametri per il provider di identità.
abilitato	bool	Indica se il provider di identità è abilitato.

Query di esempio

Per visualizzare le proprietà provider di identità, esegui una query come la seguente, dopo aver creato i provider di identità.

```
SELECT name, type, instanceid, namespc, params, enabled
FROM svv_identity_providers
```

```
ORDER BY 1;
```

L'output di esempio include le descrizioni dei parametri.

```
       name       | type  |              instanceid              | namespc |

                                        params

                      | enabled
------------------+-------+--------------------------------------+---------
+-----------------------------------------------------------------------------------------------------------------
+---------
 rs5517_azure_idp | azure | e40d4bb2-7670-44ae-bfb8-5db013221d73 | abc     |
 {"issuer":"https://login.microsoftonline.com/e40d4bb2-7670-44ae-bfb8-5db013221d73/
v2.0", "client_id":"871c010f-5e61-4fb1-83ac-98610a7e9110", "client_secret":,
 "audience":["https://analysis.windows.net/powerbi/connector/AmazonRedshift", "https://
analysis.windows.net/powerbi/connector/AWSRDS"]} | t
(1 row)
```

SVV_INTEGRATION

SVV_INTEGRATION visualizza i dettagli sulla configurazione delle integrazioni.

SVV_INTEGRATION è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni sulle integrazioni Zero-ETL, consulta Working with zero-ETL integrations (Utilizzo delle integrazioni Zero-ETL).

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
integration_id	character (128)	L'identificatore associato all'integrazione.
target_database	character (128)	Il database in Amazon Redshift che riceve i dati dell'integrazione.
source	character (128)	I dati di origine per l'integrazione. I tipi possibili sono `MySQL` e `PostgreSQL` .

Nome colonna	Tipo di dati	Descrizione
stato	character (128)	Lo stato dell'integrazione. I valori possibili includono `PendingDbConnectState` , `SchemaDiscoveryState` , `CdcRefreshState` e `ErrorState` .
current_lag	bigint	Il tempo di ritardo corrente (in millisecondi) tra l'origine e la destinazione dell'integrazione.
last_replicated_checkpoint	character (128)	L'ultimo checkpoint replicato.
total_tables_replicated	integer	Il numero totale di tabelle attualmente nello stato replicato.
total_tables_failed	integer	Il numero totale di tabelle attualmente nello stato non riuscito.
creation_time	timestamp	L'ora (UTC) in cui viene creata l'integrazione. È definita come l'ora in cui il database di destinazione viene creato dall'integrazione.

Query di esempio

Il seguente comando SQL mostra le integrazioni attualmente definite.

```
select * from svv_integration;

           integration_id              | target_database | source |      state
   | current_lag |      last_replicated_checkpoint     | total_tables_replicated |
 total_tables_failed |       creation_time
--------------------------------------+-----------------+--------+-----------------
+-------------+--------------------------------------+-------------------------
+---------------------+---------------------------
  99108e72-1cfd-414f-8cc0-0216acefac77 |     perfdb      |  MySQL | CdcRefreshState |
  56606106  | {"txn_seq":9834,"txn_id":126597515} |            152          |
 0         | 2023-09-19 21:05:27.520299
```

SVV_INTEGRATION_TABLE_STATE

SVV_INTEGRATION_TABLE_STATE visualizza i dettagli sulle informazioni sull'integrazione a livello di tabella.

SVV_INTEGRATION_TABLE_STATE è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per ulteriori informazioni, consulta Working with zero-ETL integrations (Utilizzo delle integrazioni Zero-ETL).

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
integration_id	character(128)	L'identificatore associato all'integrazione.
target_database	character(128)	Il nome del database Amazon Redshift.
schema_name	character(128)	Nome dello schema Amazon Redshift.
table_name	character(128)	Nome della tabella.
table_state	character(128)	Lo stato della tabella. I valori possibili sono `Synced`, `Failed`, `Deleted`, `ResyncRequired` e `ResyncInitiated`.
table_last_replicated_checkpoint	character(128)	Le coordinate attuali del log sincronizzato.
motivo	character(256)	Il motivo dell'ultima transizione dello stato. I motivi più comuni possono essere tipi di dati non supportati nelle tabelle o tabelle che non hanno chiavi primarie. Per ulteriori informazioni su come risolvere i problemi più comuni, consulta Troubleshooting zero-ETL integrations in Amazon Redshift (Risoluzione dei problemi relativi alle integrazioni zero-ETL in Amazon Redshift).

Nome colonna	Tipo di dati	Descrizione
last_updated_timestamp	timestamp without time zone	L'ora (UTC) dell'ultimo aggiornamento della tabella.

Query di esempio

Il seguente comando SQL mostra il log delle integrazioni.

```
select * from svv_integration_table_state;

          integration_id              | target_database | schema_name |     table_name
   | Table_state  |table_last_replicated_checkpoint | reason | last_updated_timestamp
--------------------------------------+-----------------+-------------
+------------------+-------------+--------------------------------
+--------------------------------------
 4798e675-8f9f-4686-b05f-92c538e19629 |  sample_test2   |    sample   |
 SampleTestChannel |    Synced    |   {"txn_seq":3,"txn_id":3122}   |        |
 2023-05-12 12:40:30.656625
```

SVV_INTERLEAVED_COLUMNS

Utilizzare la visualizzazione SVV_INTERLEAVED_COLUMNS per determinare se una tabella che utilizza le chiavi di ordinamento con interfoliazione debba essere reindicizzata tramite VACUUM REINDEX. Per ulteriori informazioni su come determinare la frequenza dell'esecuzione di VACUUM e quando eseguire VACUUM REINDEX, consultare Gestione dei tempi di vacuum.

SVV_INTERLEAVED_COLUMNS è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
tbl	integer	ID tabella.
col	integer	Indice in base zero della colonna.

Nome colonna	Tipo di dati	Descrizione
interleaved_skew	numeric 9,2)	Rapporto che indica il grado di inclinazione presente nelle colonne con chiavi di ordinamento con interfoliazione per una tabella. Un valore di 1,00 indica nessuna inclinazione e valori più grandi indicano più inclinazione. Le tabelle con una grande inclinazione devono essere reindicizzate con il comando VACUUM REINDEX.
last_reindex	timestar	L'ora in cui è stato eseguito l'ultimo VACUUM REINDEX per la tabella specificata. Questo valore è NULL se la tabella non è mai stata reindicizzata o se la tabella di log del sistema sottostante, STL_VACUUM, è stata ruotata dall'ultima reindicizzazione.

Query di esempio

Per identificare le tabelle che potrebbero dover essere reindicizzate, eseguire la query seguente.

```
select tbl as tbl_id, stv_tbl_perm.name as table_name,
col, interleaved_skew, last_reindex
from svv_interleaved_columns, stv_tbl_perm
where svv_interleaved_columns.tbl = stv_tbl_perm.id
and interleaved_skew is not null;

 tbl_id | table_name | col | interleaved_skew | last_reindex
--------+------------+-----+------------------+---------------------
 100068 | lineorder  |   0 |             3.65 | 2015-04-22 22:05:45
 100068 | lineorder  |   1 |             2.65 | 2015-04-22 22:05:45
 100072 | customer   |   0 |             1.65 | 2015-04-22 22:05:45
 100072 | lineorder  |   1 |             1.00 | 2015-04-22 22:05:45
(4 rows)
```

SVV_LANGUAGE_PRIVILEGES

Utilizzare SVV_LANGUAGE_PRIVILEGES per visualizzare le autorizzazioni di linguaggio concesse esplicitamente a utenti, ruoli e gruppi nel database corrente.

SVV_LANGUAGE_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati

- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
language_name	text	Il nome del linguaggio.
privilege_type	text	Il tipo di autorizzazione. Il valore possibile è USAGE.
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente, ruolo, gruppo o pubblico.
admin_option	booleano	Un valore che indica se l'utente può concedere l'autorizzazione ad altri utenti e ruoli. È sempre false per il ruolo e il tipo di identità del gruppo.

Query di esempio

L'esempio seguente mostra il risultato di SVV_LANGUAGE_PRIVILEGES.

```
SELECT language_name,privilege_type,identity_name,identity_type,admin_option FROM
 svv_language_privileges
WHERE identity_name IN ('role1', 'reguser');

 language_name | privilege_type | identity_name | identity_type | admin_option
---------------+----------------+---------------+---------------+--------------
   exfunc      |     USAGE      |    reguser    |     user      |    False
   exfunc      |     USAGE      |     role1     |     role      |    False
   plpythonu   |     USAGE      |    reguser    |     user      |    False
```

SVV_MASKING_POLICY

Utilizza SVV_MASKING_POLICY per visualizzare tutte le policy di mascheramento create nel cluster.

Solo gli utenti con privilegi avanzati e gli utenti con il ruolo `sys:secadmin` possono visualizzare SVV_MASKING_POLICY. Gli utenti con privilegi normali vedranno 0 righe.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
policy_database	text	Nome del database in cui è stata creata la policy di mascheramento.
nome_policy	text	Nome della policy di mascheramento.
input_columns	text	Gli attributi forniti nella clausola WITH dell'istruzione CREATE POLICY.
policy_expression	text	L'espressione di mascheram ento utilizzata nella policy.
policy_modified_by	text	Nome dell'utente che ha modificato per ultimo la policy.
policy_modified_time	timestamp	Il timestamp che indica il momento in cui la policy è stata creata o modificata l'ultima volta.

SVV_ML_MODEL_INFO

Indicare le informazioni sullo stato corrente del modello di machine learning.

SVV_ML_MODEL_INFO è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	char(128)	Il database del modello.
schema_name	char(128)	Lo schema del modello.
user_name	char(128)	Il proprietario del modello.
model_name	char(128)	Il nome del modello.
life_cycle	char(20)	Lo stato del ciclo di vita del modello.
is_refreshable	integer	Lo stato del modello se è aggiornabile se le tabelle e le colonne originali nella query di addestramento esistono ancora e l'utente dispone ancora delle relative autorizzazioni. I valori possibili sono: 1 (aggiornabile) e 0 (non aggiornab ile).
model_state	char(128)	Lo stato corrente del modello.

Query di esempio

La query seguente visualizza lo stato corrente dei modelli di machine learning.

```
SELECT schema_name, model_name, model_state
FROM svv_ml_model_info;

 schema_name |        model_name             |             model_state
-------------+-------------------------------+-----------------------------------
 public      | customer_churn_auto_model     | Train Model On SageMaker In Progress
 public      | customer_churn_xgboost_model | Model is Ready
(2 row)
```

SVV_ML_MODEL_PRIVILEGES

Utilizza SVV_ML_MODEL_PRIVILEGES per visualizzare le autorizzazioni del modello di machine learning concesse esplicitamente a utenti, ruoli e gruppi nel cluster.

SVV_ML_MODEL_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
namespace_name	text	Il nome dello spazio dei nomi in cui è presente il modello di machine learning specificato.
model_name	text	Il nome del modello di machine learning.
versione_modello	integer	Il numero di versione del modello.
privilege_type	text	Il tipo di autorizzazione. Il valore possibile è EXECUTE.
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente, ruolo, gruppo o pubblico.

Nome colonna	Tipo di dati	Descrizione
admin_option	booleano	Un valore che indica se l'utente può concedere l'autorizzazione ad altri utenti e ruoli. È sempre false per il ruolo e il tipo di identità del gruppo.

Query di esempio

L'esempio seguente mostra il risultato di SVV_ML_MODEL_PRIVILEGES.

```
SELECT
 namespace_name,model_name,model_version,privilege_type,identity_name,identity_type,admin_optio
 FROM svv_ml_model_privileges
WHERE model_name = 'test_model';

 namespace_name | model_name | model_version | privilege_type |  identity_name |
 identity_type | admin_option
----------------+------------+---------------+----------------+----------------
+---------------+--------------
      public    | test_model |       1       |    EXECUTE     |     reguser    |
 user      |    False
      public    | test_model |       1       |    EXECUTE     |     role1      |
 role      |    False
```

SVV_MV_DEPENDENCY

La tabella SVV_MV_DEPENDENCY mostra le dipendenze delle viste materializzate su altre viste materializzate in Amazon Redshift.

Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

SVV_MV_DEPENDENCY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	char(128)	Il database che contiene la vista materializzata specificata.
schema_name	char(128)	Lo schema della vista materializzata.
nome	char(128)	Il nome della vista materializzata.
dependent_database_name	char(128)	Il database delle viste materializzate da cui dipende la vista materializzata.
dependent_schema_name	char(128)	Lo schema delle viste materializzate da cui dipende la vista materializzata.
dependent_name	char(128)	Il nome della vista materializzata da cui dipende questa vista materializzata.

Query di esempio

La query seguente restituisce una riga di output che indica che la vista materializzata `mv_over_foo` utilizza la vista materializzata `mv_foo` nella sua definizione come dipendenza.

```
CREATE SCHEMA test_ivm_setup;
CREATE TABLE test_ivm_setup.foo(a INT);
CREATE MATERIALIZED VIEW test_ivm_setup.mv_foo AS SELECT * FROM test_ivm_setup.foo;
CREATE MATERIALIZED VIEW test_ivm_setup.mv_over_foo AS SELECT * FROM
 test_ivm_setup.mv_foo;

SELECT * FROM svv_mv_dependency;

 database_name | schema_name          | name        | dependent_database_name |
 dependent_schema_name     | dependent_name
---------------+----------------------+-------------+-------------------------
+---------------------------+----------
 dev           | test_ivm_setup       | mv_over_foo |                     dev |
 test_ivm_setup            | mv_foo
```

SVV_MV_INFO

La tabella SVV_MV_INFO contiene una riga per ogni vista materializzata, se i dati sono obsoleti, e le informazioni sullo stato.

Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

SVV_MV_INFO è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	char(128)	Il database che contiene la vista materializzata.
schema_name	char(128)	Lo schema del database.
user_name	char(128)	L'utente proprietario della vista materializzata.
nome	char(128)	Il nome della vista materializzata.
is_stale	char(1)	`t` indica che la vista materializzata è obsoleta. In una vista materializzata non aggiornata sono state aggiornate le tabelle di base ma non la vista materializzata. Queste informazioni potrebbero non essere precise se dall'ultimo riavvio non è stato eseguito alcun aggiornamento.
stato	integer	Lo stato della vista materializzata come descritto di seguito: • 0: la vista materializzata viene ricalcolata completamente quando viene aggiornata. • 1: la vista materializzata è incrementale. • 101: la vista materializzata non può essere aggiornata a causa di una colonna eliminata.

Nome colonna	Tipo di dati	Descrizione
		Questo vincolo si applica anche se la colonna non viene utilizzata nella vista materializzata. • 102: la vista materializzata non può essere aggiornata a causa di un tipo di colonna modificato. Questo vincolo si applica anche se la colonna non viene utilizzata nella vista materializzata. • 103: la vista materializzata non può essere aggiornata a causa di una tabella rinominata. • 104: la vista materializzata non può essere aggiornata a causa di una colonna rinominata. Questo vincolo si applica anche se la colonna non viene utilizzata nella vista materializzata. • 105: la vista materializzata non può essere aggiornata a causa di uno schema rinominato.
autorewrite	char(1)	Una `t` indica che la vista materializzata è idonea per la riscrittura automatica delle query.
autorefresh	char(1)	Una `t` indica che la vista materializzata può essere aggiornata automaticamente.

Query di esempio

Per visualizzare lo stato di tutte le viste materializzate, eseguire la query seguente.

```
select * from svv_mv_info;
```

Questa query restituisce l'output di esempio seguente.

```
database_name |       schema_name       | user_name |   name   |  is_stale | state |
 autorefresh | autorewrite
--------------+-------------------------+-----------+----------+-----------+-------
+-------------+----------------
```

```
dev          | test_ivm_setup          | catch-22  | mv      |   f       |     1 |
      1 |           0
dev          | test_ivm_setup          | lotr      | old_mv  |   t       |     1 |
      0 |           1
```

SVV_QUERY_INFLIGHT

Utilizzare la visualizzazione SVV_QUERY_INFLIGHT per determinare quali query sono attualmente in esecuzione sul database. Questa visualizzazione unisce STV_INFLIGHT a STL_QUERYTEXT. SVV_QUERY_INFLIGHT non mostra solo le query di nodo principale. Per ulteriori informazioni, consulta Nodo principale: solo funzioni.

SVV_QUERY_INFLIGHT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> Questa vista è disponibile solo per le query sui cluster con provisioning.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
sezione	integer	Sezione in cui viene eseguita la query.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizzazioni di sistema.
pid	integer	ID processo. Tutte le query in una sessione sono eseguite nello stesso processo, quindi questo valore rimane costante se si esegue una serie di query nella stessa sessione. È possibile utilizzare questa colonna per eseguire la connessione alla tabella STL_ERROR.

Nome colonna	Tipo di dati	Descrizione
starttime	timestamp	Ora in cui è stata avviata la query.
sospeso	integer	Se la query viene sospesa: **0** = falso; **1** = vero.
text	character(200)	Testo della query, in incrementi da 200 caratteri.
sequenza	integer	Numero di sequenza per i segmenti delle istruzioni di query.

Query di esempio

L'output di esempio qui di seguito mostra due query attualmente in esecuzione, la stessa query SVV_QUERY_INFLIGHT e la query 428, la quale è divisa in tre righe nella tabella. (In questo output di esempio, le colonne starttime e istruzione sono troncate)

```
select slice, query, pid, starttime, suspended, trim(text) as statement, sequence
from svv_query_inflight
order by query, sequence;

slice|query| pid  |      starttime       |suspended| statement | sequence
-----+-----+------+----------------------+---------+-----------+---------
1012 | 428 | 1658 | 2012-04-10 13:53:... |       0 | select ...|    0
1012 | 428 | 1658 | 2012-04-10 13:53:... |       0 | enueid ...|    1
1012 | 428 | 1658 | 2012-04-10 13:53:... |       0 | atname,...|    2
1012 | 429 | 1608 | 2012-04-10 13:53:... |       0 | select ...|    0
(4 rows)
```

SVV_QUERY_STATE

Utilizza SVV_QUERY_STATE per visualizzare le informazioni riguardanti il tempo di esecuzione delle query attualmente in esecuzione.

La visualizzazione SVV_QUERY_STATE contiene un sottoinsieme di dati della tabella STV_EXEC_STATE.

SVV_QUERY_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

> **Note**
>
> Questa vista è disponibile solo per le query sui cluster con provisioning.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizz azioni di sistema.
seg	integer	Numero del segmento di query in esecuzione. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. I segmenti di query possono essere eseguiti in parallelo. Ogni segmento viene eseguito in un singolo processo.
step	integer	Numero della fase della query in esecuzione. Una fase è l'unità più piccola del tempo di esecuzione delle query. Ogni fase rappresen ta un'unità di lavoro distinta, come per esempio la scansione di una tabella, la restituzione dei risultati o l'ordinamento dei dati.
maxtime	intervallo	Quantità di tempo massima (in microsecondi) per l'esecuzione di questa fase.
avgtime	intervallo	Tempo medio (in microsecondi) per l'esecuzione di questa fase.
righe	bigint	Numero di righe prodotte dalla fase in esecuzione.

Nome colonna	Tipo di dati	Descrizione
bytes	bigint	Numero di byte prodotti dalla fase in esecuzione.
cpu	bigint	Per uso interno.
memory	bigint	Per uso interno.
rate_row	double precision	ows-per-second Frequenza R dall'inizio della query, calcolata sommando le righe e dividendo per il numero di secondi dall'inizio della query all'ora corrente.
rate_byte	double precision	ytes-per-second Frequenza B dall'inizio della query, calcolata sommando i byte e dividendo per il numero di secondi dall'inizio della query all'ora corrente.
etichetta	character(25)	Etichetta di query: un nome per la fase, come per esempio `scan` o `sort`.
is_diskbased	character(1)	Se questa fase della query è in esecuzione come operazione basata su disco: true (**`t`**) o false (**`f`**). Solo determinate fasi, come hash, sort e le fasi di aggregazione, possono accedere al disco. Molti tipi di fase sono sempre eseguiti in memoria.
workmem	bigint	Quantità di memoria di lavoro (in byte) assegnata alla fase di query.
num_parts	integer	Numero di partizioni in cui una tabella di hash è divisa durante una fase di hash. Un numero positivo in questa colonna non implica che la fase di hash viene eseguita come operazione basata su disco. Verifica il valore nella colonna IS_DISKBASED per determinare se la fase di hash era basata sul disco.
is_rrscan	character(1)	Se true (**`t`**), indica che in questa fase è stata utilizzata la scansione a intervallo limitato. Il valore predefinito è false (**`f`**).
is_delayed_scan	character(1)	Se true (**`t`**), indica che in questa fase è stata utilizzata la scansione ritardata. Il valore predefinito è false (**`f`**).

Query di esempio

Determinazione del tempo di elaborazione di una query in base alla fase

La query seguente mostra quanto è durata l'esecuzione di ogni fase della query con l'ID query 279 e quante righe di dati sono state elaborate da Amazon Redshift:

```
select query, seg, step, maxtime, avgtime, rows, label
from svv_query_state
where query = 279
order by query, seg, step;
```

Questa query recupera le informazioni di elaborazione relative alla query 279, come mostrato nel seguente output di esempio:

```
query |   seg   | step | maxtime | avgtime |  rows   | label
------+---------+------+---------+---------+---------+------------------
  279 |       3 |    0 | 1658054 | 1645711 | 1405360 | scan
  279 |       3 |    1 | 1658072 | 1645809 |       0 | project
  279 |       3 |    2 | 1658074 | 1645812 | 1405434 | insert
  279 |       3 |    3 | 1658080 | 1645816 | 1405437 | distribute
  279 |       4 |    0 | 1677443 | 1666189 | 1268431 | scan
  279 |       4 |    1 | 1677446 | 1666192 | 1268434 | insert
  279 |       4 |    2 | 1677451 | 1666195 |       0 | aggr
(7 rows)
```

Determinazione di eventuali query attive attualmente in esecuzione sul disco

La query seguente mostra la presenza di eventuali query attive attualmente in esecuzione sul disco:

```
select query, label, is_diskbased from svv_query_state
where is_diskbased = 't';
```

Questo output di esempio mostra la presenza di eventuali query attive attualmente in esecuzione sul disco:

```
 query | label        | is_diskbased
-------+--------------+--------------
1025   | hash tbl=142 |      t
(1 row)
```

SVV_REDSHIFT_COLUMNS

Utilizza SVV_REDSHIFT_COLUMNS per visualizzare un elenco di tutte le colonne a cui un utente ha accesso. Questo insieme di colonne include le colonne del cluster e le colonne delle unità di condivisione dati fornite dai cluster remoti.

SVV_REDSHIFT_COLUMNS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	varchar(128)	Il nome del database in cui esiste la tabella contenente le colonne.
schema_name	varchar(128)	Il nome dello schema per la tabella.
table_name	varchar(128)	Nome della tabella.
column_name	varchar(128)	Il nome di una colonna.
ordinal_position	integer	La posizione della colonna nella tabella.
data_type	varchar(32)	Il tipo di dati della colonna.
column_default	varchar(4000)	Il valore predefinito della colonna.
is_nullable	varchar(3)	Un valore che Indica se la colonna è nullable. I valori possibili sono `yes`, no o " " (una stringa vuota che rappresenta nessuna informazione).

Nome colonna	Tipo di dati	Descrizione
encoding	varchar(128)	Il tipo di codifica della colonna.
distkey	booleano	Un valore che è true se la colonna corrisponde alla chiave di distribuzione per la tabella e false in caso contrario.
sortkey	integer	Un valore che specifica l'ordine della colonna nella chiave di ordinamento. Se la tabella usa una chiave di ordinamento composta, tutte le colonne che fanno parte della chiave di ordinamen to hanno un valore positivo che indica la posizione della colonna nella chiave di ordinamento. Se la tabella usa una chiave di ordinamento interleav ed, allora ogni colonna che fa parte della chiave di ordinamento ha un valore alternativamente positivo o negativo. Qui, il valore assoluto indica la posizione della colonna nella chiave di ordinamento. Se `sortkey` è 0, la colonna non fa parte di una chiave di ordinamento.

Nome colonna	Tipo di dati	Descrizione
column_acl	varchar(128)	Una stringa che definisce le autorizzazioni per l'utente o il gruppo di utenti specificato per la colonna.
osservazioni	varchar(256)	Osservazioni.

Query di esempio

L'esempio seguente restituisce l'output di SVV_REDSHIFT_COLUMNS.

```
SELECT *
FROM svv_redshift_columns
WHERE database_name = 'tickit_db'
    AND TABLE_NAME = 'tickit_sales_redshift'
ORDER BY COLUMN_NAME,
    TABLE_NAME,
    database_name
LIMIT 5;

database_name | schema_name |       table_name      | column_name | ordinal_position |
 data_type | column_default | is_nullable | encoding | distkey | sortkey | column_acl
 | remarks
--------------+-------------+-----------------------+-------------+------------------
+-----------+----------------+-------------+----------+---------+---------
+-------------+--------
   tickit_db  |   public    | tickit_sales_redshift |   buyerid   |        4         |
 integer  |                |      NO     |   az64   |  False  |    0    |             |
   tickit_db  |   public    | tickit_sales_redshift |  commission |        9         |
 numeric  |      (8,2)     |     YES     |   az64   |  False  |    0    |             |
   tickit_db  |   public    | tickit_sales_redshift |    dateid   |        6         |
 smallint |                |      NO     |   none   |  False  |    1    |             |
   tickit_db  |   public    | tickit_sales_redshift |   eventid   |        5         |
 integer  |                |      NO     |   az64   |  False  |    0    |       |
   tickit_db  |   public    | tickit_sales_redshift |   listid    |        2         |
 integer  |                |      NO     |   az64   |  True   |    0    |             |
```

SVV_REDSHIFT_DATABASE

Utilizza SVV_REDSHIFT_DATABASES per visualizzare un elenco di tutti i database a cui un utente ha accesso. Ciò include i database nel cluster e i database creati dalle unità di condivisione dati fornite da cluster remoti.

SVV_REDSHIFT_DATABASES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	varchar(128)	Nome del database.
database_owner	integer	L'ID utente del proprietario del database.
database_type	varchar(32)	Il tipo di database. I tipi possibili sono database locali o condivisi.
database_acl	varchar(128)	Queste informazioni sono solo per uso interno.
database_options	varchar(128)	Le proprietà del database.
database_isolation_level	varchar(128)	Il livello di isolamento del database. I valori possibili sono `Snapshot Isolation` e `Serializable` .

Query di esempio

L'esempio seguente restituisce l'output di SVV_REDSHIFT_DATABASES.

```
select database_name, database_owner, database_type, database_options,
 database_isolation_level
```

```
from  svv_redshift_databases;


database_name | database_owner | database_type | database_options |
 database_isolation_level
--------------+----------------+---------------+------------------+------------------
   dev        |  1             | local         | NULL             | Serializable
```

SVV_REDSHIFT_FUNCTIONS

Utilizza SVV_REDSHIFT_FUNCTIONS per visualizzare un elenco di tutte le funzioni a cui un utente ha accesso. Questo insieme di funzioni include le funzioni del cluster e le funzioni delle unità di condivisione dati fornite dai cluster remoti.

SVV_REDSHIFT_FUNCTIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	varchar(128)	Il nome del database in cui è presente il cluster che dispone di queste funzioni.
schema_name	varchar(128)	Il nome dello schema che specifica una determinata funzione.
function_name	varchar(128)	Il nome di una funzione specificata.
function_type	varchar(128)	Il tipo di funzione. I valori possibili sono funzioni regolari, funzioni di aggregazione e procedure archiviate.

Nome colonna	Tipo di dati	Descrizione
argument_type	varchar(512)	Una stringa che rappresenta il tipo di argomento di input di una funzione.
result_type	varchar(128)	Il tipo di dati del valore restituito di una funzione.

Query di esempio

L'esempio seguente restituisce l'output di SVV_REDSHIFT_FUNCTIONS.

```
SELECT *
FROM svv_redshift_functions
WHERE database_name = 'tickit_db'
    AND SCHEMA_NAME = 'public'
ORDER BY function_name
LIMIT 5;

database_name | schema_name |      function_name    |  function_type   |
 argument_type  | result_type
--------------+-------------+-----------------------+------------------
+------------------+-------------
   tickit_db  |    public   |     shared_function   | REGULAR FUNCTION | integer,
 integer |   integer
```

SVV_REDSHIFT_SCHEMA_QUOTA

Visualizza la quota e l'utilizzo corrente del disco per ogni schema in un database.

SVV_REDSHIFT_SCHEMA_QUOTA è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Questa visualizzazione è disponibile per le query su cluster con provisioning o su gruppi di lavoro Redshift Serverless.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	character(128)	Il database che contiene lo schema.
schema_name	character(128)	Il nome dello schema.
schema_owner	integer	L'ID utente interno del proprietario dello schema.
quota	integer	La quantità di spazio su disco (in MB) che lo schema può utilizzare.
disk_usage	integer	Lo spazio su disco (in MB) attualmente utilizzato dallo schema.

Query di esempio

Nell'esempio seguente viene visualizzata la quota e l'utilizzo corrente del disco per lo schema denominato `sales_schema`.

```
SELECT TRIM(SCHEMA_NAME) "schema_name", QUOTA, disk_usage FROM
 svv_redshift_schema_quota
WHERE SCHEMA_NAME = 'sales_schema';


schema_name   | quota | disk_usage
--------------+-------+-----------
sales_schema  | 2048  | 30
```

SVV_REDSHIFT_SCHEMAS

Utilizza SVV_REDSHIFT_SCHEMAS per visualizzare un elenco di tutti gli schemi a cui un utente ha accesso. Questo insieme di schemi include gli schemi del cluster e gli schemi delle unità di condivisione dati fornite dai cluster remoti.

SVV_REDSHIFT_SCHEMAS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	varchar(128)	Il nome del database in cui è presente lo schema specifica to.
schema_name	varchar(128)	Lo spazio dei nomi o il nome dello schema.
schema_owner	integer	L'ID utente interno del proprietario dello schema.
schema_type	varchar(16)	Il tipo di schema. I valori possibili sono schemi condivisi e locali.
schema_acl	varchar(128)	La stringa che definisce le autorizzazioni per l'utente o il gruppo di utenti specificato per lo schema.
schema_option	varchar(128)	Le opzioni dello schema.

Query di esempio

L'esempio seguente restituisce l'output di SVV_REDSHIFT_SCHEMAS.

```
SELECT *
FROM svv_redshift_schemas
WHERE database_name = 'tickit_db'
ORDER BY database_name,
    SCHEMA_NAME;
```

```
database_name |    schema_name     | schema_owner | schema_type | schema_acl |
 schema_option
--------------+--------------------+--------------+-------------+------------
+---------------
   tickit_db  |       public       |       1      |    shared   |            |
```

SVV_REDSHIFT_TABLES

Utilizza SVV_REDSHIFT_TABLES per visualizzare un elenco di tutte le tabelle a cui un utente ha accesso. Questo insieme di tabelle include le tabelle del cluster e le tabelle delle unità di condivisione dati fornite dai cluster remoti.

SVV_REDSHIFT_TABLES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database_name	varchar(128)	Il nome del database in cui è presente la tabella specificata.
schema_name	varchar(128)	Il nome dello schema per la tabella.
table_name	varchar(128)	Nome della tabella.
table_type	varchar(128)	Il tipo di tabella. I valori possibili sono viste e tabelle.
table_acl	varchar(128)	La stringa che definisce le autorizzazioni per l'utente o il gruppo di utenti specificato per la tabella.
osservazioni	varchar(128)	Osservazioni.
table_owner	varchar(128)	Il proprietario della tabella.

Query di esempio

L'esempio seguente restituisce l'output di SVV_REDSHIFT_TABLES.

```
SELECT *
FROM svv_redshift_tables
WHERE database_name = 'tickit_db' AND TABLE_NAME LIKE 'tickit_%'
ORDER BY database_name,
TABLE_NAME;

database_name | schema_name |         table_name       | table_type | table_acl |
 remarks | table_owner
--------------+-------------+--------------------------+------------+-----------
+---------+-----------
   tickit_db  |    public   | tickit_category_redshift |    TABLE   |           |
   +
   tickit_db  |    public   |   tickit_date_redshift   |    TABLE   |           |
   +
   tickit_db  |    public   |   tickit_event_redshift  |    TABLE   |           |
   +
   tickit_db  |    public   |  tickit_listing_redshift |    TABLE   |           |
   +
   tickit_db  |    public   |   tickit_sales_redshift  |    TABLE   |           |
   +
   tickit_db  |    public   |   tickit_users_redshift  |    TABLE   |           |
   +
   tickit_db  |    public   |   tickit_venue_redshift  |    TABLE   |           |
```

Se il valore table_acl è nullo, non sono stati concessi esplicitamente privilegi di accesso alla tabella corrispondente.

SVV_RELATION_PRIVILEGES

Utilizza SVV_RELATION_PRIVILEGES per visualizzare le autorizzazioni di relazione (tabelle e viste) concesse esplicitamente a utenti, ruoli e gruppi nel database corrente.

SVV_RELATION_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione SYSLOG ACCESS UNRESTRICTED

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari. Per ulteriori informazioni sulla visibilità dei dati, vedere. Visibilità dei dati nelle tabelle e nelle viste di sistema

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
namespace _name	text	Il nome dello spazio dei nomi in cui è presente la relazione specificata.
relation_name	text	Il nome della relazione.
privilege_type	text	Il tipo di autorizzazione. I valori possibili sono INSERT, SELECT, UPDATE, DELETE, REFERENCES o DROP.
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente, ruolo, gruppo o pubblico.
admin_option	booleano	Un valore che indica se l'utente può concedere l'autoriz zazione ad altri utenti e ruoli. È sempre false per il ruolo e il tipo di identità del gruppo.

Query di esempio

L'esempio seguente mostra il risultato di SVV_RELATION_PRIVILEGES.

```
SELECT
 namespace_name,relation_name,privilege_type,identity_name,identity_type,admin_option
 FROM svv_relation_privileges
WHERE relation_name = 'orders' AND privilege_type = 'SELECT';

 namespace_name | relation_name | privilege_type |  identity_name | identity_type |
 admin_option
```

```
---------------+--------------+---------------+---------------+--------------
+-------------
     public     |    orders     |     SELECT     |    reguser     |     user      |
 False
     public     |    orders     |     SELECT     |     role1      |     role      |
 False
```

SVV_RLS_APPLIED_POLICY

Utilizza SVV_RLS_APPLIED_POLICY per tracciare l'applicazione delle policy RLS su query che fanno riferimento a relazioni protette da RLS.

SVV_RLS_APPLIED_POLICY è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con il ruolo `sys:operator`
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Tieni presente che a sys:secadmin non è concessa questa autorizzazione di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
username	text	Il nome dell'utente che ha eseguito la query.
query	integer	L'ID della query.
xid	Long	Il contesto della transazione.
pid	integer	Il processo principale che esegue la query.
recordtime	time	L'ora in cui è stata registrata la query.
command	char(1)	Il comando per il quale è stata applicata la policy RLS. I valori possibili sono k (sconosciuto), `s` (selezione), u (aggiornamento), `i` (inserimento), y (utilità) e d (eliminazione).

Nome colonna	Tipo di dati	Descrizione
datname	text	Il nome del database della relazione a cui è allegata la policy di sicurezza a livello di riga.
relschema	text	Il nome dello schema della relazione a cui è allegata la policy di sicurezza a livello di riga.
relname	text	Il nome della relazione a cui è allegata la policy di sicurezza a livello di riga.
polname	text	Il nome della policy di sicurezza a livello di riga collegata alla relazione.
poldefault	char(1)	L'impostazione predefinita della policy di sicurezza a livello di riga collegata alla relazione. I valori possibili sono `f` per false se è stata applicata la policy false predefinita e `t` per true se è stata applicata la policy true predefinita.

Query di esempio

L'esempio seguente mostra il risultato di SVV_RLS_APPLIED_POLICY. Per interrogare SVV_RLS_APPLIED_POLICY, è necessaria l'autorizzazione ACCESS SYSTEM TABLE.

```
-- Check what RLS policies were applied to the run query.
SELECT username, command, datname, relschema, relname, polname, poldefault
FROM svv_rls_applied_policy
WHERE datname = CURRENT_DATABASE() AND query = PG_LAST_QUERY_ID();

 username | command |  datname  | relschema |          relname         |      polname
  | poldefault
----------+---------+-----------+-----------+--------------------------
+-----------------+------------
   molly  |    s    | tickit_db |   public  | tickit_category_redshift |
 policy_concerts |
```

SVV_RLS_ATTACHED_POLICY

Utilizza SVV_RLS_ATTACHED_POLICY per visualizzare un elenco di tutte le relazioni e gli utenti che hanno una o più policy di sicurezza a livello di riga collegate al database attualmente connesso.

Solo gli utenti con il ruolo sys:secadmin possono interrogare questa vista.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
relschema	text	Il nome dello schema della relazione a cui è allegata la policy di sicurezza a livello di riga.
relname	text	Il nome della relazione a cui è allegata la policy di sicurezza a livello di riga.
relkind	text	Il tipo di oggetto, ad esempio una tabella.
polname	text	Il nome della policy di sicurezza a livello di riga collegata alla relazione.
grantor	text	Il nome dell'utente che ha allegato questa policy.
grantee	text	Il nome dell'utente o del ruolo a cui è allegata questa policy.
granteekind	text	Il tipo di assegnatario. I valori possibili sono utente o ruolo.
is_pol_on	booleano	Il parametro che indica se una policy di sicurezza a livello di riga è attivata o disattivata su una tabella. I valori possibili sono true e false.
is_rls_on	booleano	Il parametro che indica se una sicurezza a livello di riga è attivata o disattivata su una tabella. I valori possibili sono true e false.
rls_conjunction_type	character (3)	Il parametro che indica se la relazione combina le policy RLS con `and` o `or`.

Query di esempio

L'esempio seguente mostra il risultato di SVV_RLS_ATTACHED_POLICY.

```
--Inspect the policy in SVV_RLS_ATTACHED_POLICY
SELECT * FROM svv_rls_attached_policy;

 relschema |        relname           | relkind |     polname     | grantor | grantee
 | granteekind | is_pol_on | is_rls_on | rls_conjuntion_type
-----------+--------------------------+---------+-----------------+---------+----------
+-------------+-----------+-----------+---------------------
 public    | tickit_category_redshift |  table  | policy_concerts |   bob   |  analyst
 |    role     |    True   |    True   |      and
 public    | tickit_category_redshift |  table  | policy_concerts |   bob   |  dbadmin
 |    role     |    True   |    True   |      and
```

SVV_RLS_POLICY

Utilizza SVV_RLS_POLICY per visualizzare un elenco di tutte le policy di sicurezza a livello di riga create nel cluster Amazon Redshift.

SVV_RLS_POLICY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
poldb	text	Il nome del database in cui viene creata la policy di sicurezza a livello di riga.
polname	text	Il nome della policy di sicurezza a livello di riga.
polalias	text	L'alias della tabella utilizzato nella definizione di policy.
polatts	text	Gli attributi forniti per la definizione della policy.
polqual	text	La condizione della policy fornita nella clausola USING dell'istruzione CREATE POLICY.

Nome colonna	Tipo di dati	Descrizione
polenabled	booleano	Indica se la policy viene attivata a livello globale.
polmodifiedby	text	Il nome dell'utente che ha creato o modificato la policy più di recente.
polmodifiedtime	timestamp	Il timestamp che indica il momento in cui la policy è stata creata o modificata l'ultima volta.

Query di esempio

L'esempio seguente mostra il risultato di SVV_RLS_POLICY.

```
-- Create some policies.
CREATE RLS POLICY pol1 WITH (a int) AS t USING ( t.a IS NOT NULL );
CREATE RLS POLICY pol2 WITH (c varchar(10)) AS t USING ( c LIKE '%public%');

-- Inspect the policy in SVV_RLS_POLICY
SELECT * FROM svv_rls_policy;

 poldb | polname | polalias |                    polatts                     |
         polqual               | polenabled | polmodifiedby |   polmodifiedtime
-------+---------+----------+------------------------------------------------
+-------------------------------------+------------+--------------
+---------------------
 my_db | pol1    | t        | [{"colname":"a","type":"integer"}]             |
 "t"."a" IS NOT NULL                  | t          | policy_admin  | 2022-02-11
 14:40:49
 my_db | pol2    | t        | [{"colname":"c","type":"character varying(10)"}] |
 "t"."c" LIKE CAST('%public%' AS TEXT) | t          | policy_admin  | 2022-02-11
 14:41:28
```

SVV_RLS_RELATION

Utilizza SVV_RLS_RELATION per visualizzare un elenco di tutte le relazioni protette da RLS.

SVV_RLS_RELATION è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
datname	text	Il nome del database contenente la relazione.
relschema	text	Il nome dello schema contenente la relazione.
relname	text	Il nome della relazione.
relkind	text	Il tipo di relazione, ad esempio tabelle o viste.
is_rls_on	booleano	Il parametro che indica se la relazione è protetta da RLS.
is_rls_datashare_on	booleano	Il parametro che indica se la relazione è protetta da RLS (sicurezza a livello di riga) nelle unità di condivisione dati.
rls_conjunction_type	character(3)	Il parametro che indica se la relazione combina le policy RLS con `and` o `or`.
rls_datashare_conjunction_type	character(3)	Il parametro che indica se la relazione combina le policy RLS con `and` o `or` nelle unità di condivisione dati.

Query di esempio

L'esempio seguente mostra il risultato di SVV_RLS_RELATION.

```
ALTER TABLE tickit_category_redshift ROW LEVEL SECURITY ON FOR DATASHARES;


--Inspect RLS state on the relations using SVV_RLS_RELATION.
SELECT datname, relschema, relname, relkind, is_rls_on, is_rls_datashare_on FROM
 svv_rls_relation ORDER BY relname;

  datname  | relschema |        relname           | relkind | is_rls_on |
 is_rls_datashare_on | rls_conjunction_type | rls_datashare_conjunction_type
-----------+-----------+--------------------------+---------+-----------
+---------------------+----------------------+--------------------------------
```

```
 tickit_db |   public  | tickit_category_redshift |  table  |      t    |          t
       |          and         |              and
(1 row)
```

SVV_ROLE_GRANTS

Utilizza SVV_ROLE_GRANTS per visualizzare un elenco di ruoli a cui sono esplicitamente concessi ruoli nel cluster.

SVV_ROLE_GRANTS è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
role_id	integer	L'ID del ruolo.
role_name	text	Il nome del ruolo.
granted_role_id	integer	L'ID del ruolo concesso.
granted_role_name	text	Il nome del ruolo concesso.

Query di esempio

L'esempio seguente restituisce l'output di SVV_ROLE_GRANTS.

```
GRANT ROLE role1 TO ROLE role2;
GRANT ROLE role2 TO ROLE role3;

SELECT role_name, granted_role_name FROM svv_role_grants;

 role_name |  granted_role_name
```

```
-----------+--------------------
   role2   |      role1
   role3   |      role2
(2 rows)
```

SVV_ROLES

Utilizza SVV_ROLES per visualizzare un elenco di ruoli a cui un utente ha accesso.

SVV_ROLES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
role_id	integer	L'ID ruolo.
role_name	text	Il nome del ruolo.
role_owner	text	Il nome del proprietario del ruolo.
external_id	text	L'identificatore univoco del ruolo nel provider di identità di terze parti.

Query di esempio

L'esempio seguente restituisce l'output di SVV_ROLES.

```
SELECT role_name,role_owner FROM svv_roles WHERE role_name IN ('role1', 'role2');

 role_name | role_owner
-----------+------------
   role1   | superuser
```

```
role2   | superuser
```

SVV_SCHEMA_PRIVILEGES

Utilizza SVV_SCHEMA_PRIVILEGES per visualizzare le autorizzazioni di schema concesse esplicitamente a utenti, ruoli e gruppi nel database corrente.

SVV_SCHEMA_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
namespace_name	text	Il nome dello spazio dei nomi in cui è presente lo schema specificato.
privilege_type	text	Il tipo di autorizzazione. I valori possibili sono USAGE o CREATE.
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente, ID ruolo o ID gruppo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente, ruolo, gruppo o pubblico.
admin_option	booleano	Un valore che indica se l'utente può concedere l'autorizzazione ad altri utenti e ruoli. È sempre false per il ruolo e il tipo di identità del gruppo.

Query di esempio

L'esempio seguente mostra il risultato di SVV_SCHEMA_PRIVILEGES.

```
SELECT namespace_name,privilege_type,identity_name,identity_type,admin_option FROM
 svv_schema_privileges
WHERE namespace_name = 'test_schema1';

 namespace_name | privilege_type |  identity_name | identity_type | admin_option
----------------+----------------+----------------+---------------+--------------
 test_schema1   |    USAGE       |     reguser    |     user      |   False
 test_schema1   |    USAGE       |     role1      |     role      |   False
```

SVV_SCHEMA_QUOTA_STATE

Visualizza la quota e l'utilizzo corrente del disco per ogni schema.

Gli utenti normali possono visualizzare le informazioni relative agli schemi per i quali hanno l'autorizzazione di utilizzo. Gli utenti con privilegi avanzati possono visualizzare le informazioni per tutti gli schemi nel database corrente.

SVV_SCHEMA_QUOTA_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> Questa vista è disponibile solo per le query sui cluster con provisioning.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
schema_id	integer	Lo spazio dei nomi o l'ID dello schema.
schema_name	carattere (128)	Lo spazio dei nomi o il nome dello schema.

Nome colonna	Tipo di dati	Descrizione
schema_owner	integer	L'ID utente interno del proprietario dello schema.
quota	integer	La quantità di spazio su disco (in MB) che lo schema può utilizzare.
disk_usage	integer	Lo spazio su disco (in MB) attualmente utilizzato dallo schema.
disk_usage_pct	double precision	Percentuale di spazio su disco attualmente utilizzata dallo schema al di fuori dalla quota configurata.

Query di esempio

Nell'esempio seguente viene visualizzata la quota e l'utilizzo corrente del disco per lo schema.

```
SELECT TRIM(SCHEMA_NAME) "schema_name", QUOTA, disk_usage, disk_usage_pct FROM
 svv_schema_quota_state
WHERE SCHEMA_NAME = 'sales_schema';
schema_name   | quota | disk_usage | disk_usage_pct
--------------+-------+------------+----------------
sales_schema  | 2048  | 30         | 1.46
(1 row)
```

SVV_SYSTEM_PRIVILEGES

SVV_SYSTEM_PRIVILEGES è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo le identità a cui hanno accesso o di cui sono proprietari.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
system_privilege	text	Il nome delle autorizzazioni di sistema.
identity_id	integer	L'ID dell'identità. I valori possibili sono ID utente o ID ruolo.
identity_name	text	Il nome dell'identità.
identity_type	text	Tipo dell'identità. I valori possibili sono utente o ruolo.

Query di esempio

L'esempio seguente mostra i risultati dei parametri specificati.

```
SELECT system_privilege,identity_name,identity_type FROM svv_system_privileges
WHERE system_privilege = 'ALTER TABLE' AND identity_name = 'sys:superuser';

 system_privilege | identity_name | identity_type
------------------+---------------+--------------
   ALTER TABLE    | sys:superuser |     role
```

SVV_TABLE_INFO

Mostra le informazioni di riepilogo per le tabelle nel database. La visualizzazione filtra le tabelle di sistema e mostra solo quelle definite dall'utente.

Puoi utilizzare la visualizzazione SVV_TABLE_INFO per diagnosticare e risolvere problemi di progettazione delle tabelle che possono influenzare le prestazioni della query. Sono inclusi problemi di codifica di compressione, chiavi di distribuzione, stile di ordinamento, differenza di distribuzione dei dati, dimensioni delle tabelle e statistiche. La visualizzazione SVV_TABLE_INFO non restituisce alcuna informazione per le tabelle vuote.

La vista SVV_TABLE_INFO riepiloga le informazioni dalle tabelle di sistema STV_BLOCKLIST, STV_NODE_STORAGE_CAPACITY, STV_TBL_PERM e STV_SLICES e dalle tabelle di catalogo PG_DATABASE, PG_ATTRIBUTE, PG_CLASS, PG_NAMESPACE e PG_TYPE.

SVV_TABLE_INFO è visibile solo per gli utenti con privilegi avanzati. Per maggiori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema. Per consentire a un utente di eseguire una query sulla vista, concedi l'autorizzazione SELECT su SVV_TABLE_INFO all'utente.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database	text	Nome del database.
schema	text	Nome schema.
table_id	oid	ID tabella.
table	text	Nome tabella.
encoded	text	Valore che indica se ci sono colonne con codifica di compressione definita.
diststyle	text	Colonna dello stile di distribuzione o della chiave di distribuzione, se la chiave di distribuzione è definita. I valori possibili sono EVEN, KEY(*column*), ALL, AUTO(ALL) , AUTO(EVEN) e AUTO(KEY(*column*)).
sortkey1	text	Prima colonna nella chiave di ordinamento, se è definita una chiave di ordinamento. I valori possibili sono *column*, AUTO(SORTKEY) e AUTO(SORTKEY(*column*)).

Nome colonna	Tipo di dati	Descrizione
max_varchar	integer	Dimensione della colonna più grande che utilizza un tipo di dati VARCHAR.
sortkey1_enc	character(32)	Codifica di compressione della prima colonna nella chiave di ordinamento, se è definita una chiave di ordinamento.
sortkey_num	integer	Numero di colonne definite come chiavi di ordinamento.
size	bigint	Dimensione della tabella, in blocchi di dati da 1 MB.
pct_used	numeric(10,4)	Percentuale di spazio disponibile utilizzato dalla tabella.
empty	bigint	Per uso interno. Questa colonna non è più utilizzata e verrà rimossa in una versione futura.
unsorted	numeric(5,2)	Percentuale di righe non ordinate nella tabella.
stats_off	numeric(5,2)	Numero che indica di quanto non sono aggiornate le statistiche della tabella; 0 è attuale, 100 è non aggiornato.
tbl_rows	numeric(38,0)	Numero totale di righe nella tabella. Questo valore include le righe contrassegnate per l'eliminazione, ma non ancora sottoposte a vacuum.

Nome colonna	Tipo di dati	Descrizione
skew_sortkey1	numeric(19,2)	Rapporto tra la dimension e della colonna più grande della chiave di non ordinamen to e la dimensione della prima colonna della chiave di ordinamento, se è definita una chiave di ordinamento. Utilizzare questo valore per valutare l'efficacia della chiave di ordinamento.
skew_rows	numeric(19,2)	Rapporto tra il numero di righe della sezione con il maggior numero di righe e il numero di righe della sezione con il minor numero di righe.
estimated_visible_rows	numeric(38,0)	Le righe stimate nella tabella. In questo valore non sono incluse le righe contrasse gnate per l'eliminazione.

Nome colonna	Tipo di dati	Descrizione

risk_event	text	Informazioni sui rischi di una tabella. Il campo è separato in due parti:
		`risk_type |xid|timestamp`
		• Il `risk_type` , dove `1` indica che è stato eseguito un `COPY command with the EXPLICIT_IDS option`. Amazon Redshift non verifica più l'univocità delle colonne IDENTITY nella tabella. Per ulteriori informazioni, consulta EXPLICIT_IDS.
		• ID della transazione, `xid`, che introduce il rischio.
		• `timestamp` quando viene eseguito il comando COPY.
		L'esempio seguente mostra i valori nel campo.
		`1|1107|2019-06-22 07:16:11.292952`

vacuum_sort_benefit	numerico (12,2)	Il miglioramento percentuale massimo stimato delle prestazioni della query di scansione quando dopo l'esecuzione dell'ordinamento vacuum.

Nome colonna	Tipo di dati	Descrizione
create_time	timestamp without time zone	Timestamp relativo al momento della creazione della tabella.

Query di esempio

L'esempio seguente mostra la codifica, lo stile di distribuzione, l'ordinamento e la differenza di dati per tutte le tabelle definite dall'utente nel database. Il termine "table" qui deve essere racchiuso tra virgolette doppie perché è una parola riservata.

```
select "table", encoded, diststyle, sortkey1, skew_sortkey1, skew_rows
from svv_table_info
order by 1;

table          | encoded | diststyle       | sortkey1     | skew_sortkey1 | skew_rows
---------------+---------+-----------------+--------------+---------------+----------
category       | N       | EVEN            |              |               |
date           | N       | ALL             | dateid       |          1.00 |
event          | Y       | KEY(eventid)    | dateid       |          1.00 |      1.02
listing        | Y       | KEY(listid)     | dateid       |          1.00 |      1.01
sales          | Y       | KEY(listid)     | dateid       |          1.00 |      1.02
users          | Y       | KEY(userid)     | userid       |          1.00 |      1.01
venue          | N       | ALL             | venueid      |          1.00 |
(7 rows)
```

SVV_TABLES

Utilizzare SVV_TABLES per visualizzare le tabelle nei cataloghi locali ed esterni.

SVV_TABLES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
table_catalog	text	Il nome del catalogo in cui vi è la tabella.
table_schema	text	Il nome dello schema per la tabella.
table_name	text	Nome della tabella.
table_type	text	Il tipo di tabella. I valori possibili sono viste, tabelle esterne e tabelle di base.
osservazioni	text	Osservazioni.

SVV_TRANSACTIONS

Registra le informazioni sulle transazioni che attualmente contengono blocchi sulle tabelle del database. Utilizza la visualizzazione SVV_TRANSACTIONS per identificare le transazioni aperte e i conflitti di blocco. Per ulteriori informazioni sui blocchi, consultare Gestione delle operazioni di scrittura simultanee e LOCK.

SVV_TRANSACTIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
txn_owner	text	Nome del proprietario della transazione.
txn_db	text	Nome del database associato alla transazione.

Nome colonna	Tipo di dati	Descrizione
xid	bigint	ID transazione.
pid	integer	ID di processo associato al blocco.
txn_start	timestamp	Ora di inizio della transazione.
lock_mode	text	Nome della modalità di blocco posseduta o richiesta da questo processo. Se `lock_mode` è `Exclusive Lock` ed `granted` è vero (`t`), allora questo ID di transazione è una transazione aperta.
lockable_object_type	text	Tipo di oggetto che richiede o possiede il blocco, `relation` se è una tabella o `transactionid` se è una transazione.
relation	integer	ID di tabella per la tabella (relazione) che acquisisce il blocco. Questo valore è NULL se `lockable_object_type` è `transactionid` .
granted	booleano	Valore che indica se il blocco è stato concesso (`t`) o è in attesa (`f`).

Query di esempio

Il comando seguente mostra tutte le transazioni attive e i blocchi richiesti da ogni transazione.

```
select * from svv_transactions;
```

```
 txn_
 lockable_
 owner | txn_db |  xid   |  pid  |         txn_start          |      lock_mode      |
 object_type    | relation | granted
-------+--------+--------+-------+----------------------------+---------------------
+----------------+----------+---------
 root  | dev    | 438484 | 22223 | 2016-03-02 18:42:18.862254 | AccessShareLock     |
 relation       |   100068 | t
 root  | dev    | 438484 | 22223 | 2016-03-02 18:42:18.862254 | ExclusiveLock       |
 transactionid  |          | t
 root  | tickit | 438490 | 22277 | 2016-03-02 18:42:48.084037 | AccessShareLock     |
 relation       |    50860 | t
 root  | tickit | 438490 | 22277 | 2016-03-02 18:42:48.084037 | AccessShareLock     |
 relation       |    52310 | t
 root  | tickit | 438490 | 22277 | 2016-03-02 18:42:48.084037 | ExclusiveLock       |
 transactionid  |          | t
 root  | dev    | 438505 | 22378 | 2016-03-02 18:43:27.611292 | AccessExclusiveLock |
 relation       |   100068 | f
 root  | dev    | 438505 | 22378 | 2016-03-02 18:43:27.611292 | RowExclusiveLock    |
 relation       |    16688 | t
 root  | dev    | 438505 | 22378 | 2016-03-02 18:43:27.611292 | AccessShareLock     |
 relation       |   100064 | t
 root  | dev    | 438505 | 22378 | 2016-03-02 18:43:27.611292 | AccessExclusiveLock |
 relation       |   100166 | t
 root  | dev    | 438505 | 22378 | 2016-03-02 18:43:27.611292 | AccessExclusiveLock |
 relation       |   100171 | t
 root  | dev    | 438505 | 22378 | 2016-03-02 18:43:27.611292 | AccessExclusiveLock |
 relation       |   100190 | t
 root  | dev    | 438505 | 22378 | 2016-03-02 18:43:27.611292 | ExclusiveLock       |
 transactionid  |          | t
(12 rows)

(12 rows)
```

SVV_USER_GRANTS

Utilizza SVV_USER_GRANTS per visualizzare l'elenco degli utenti a cui sono esplicitamente concessi ruoli nel cluster.

SVV_USER_GRANTS è visibile ai seguenti utenti:

- Utenti con privilegi avanzati

- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli altri utenti possono vedere solo i ruoli esplicitamente assegnati loro.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	l'ID utente dell'utente.
user_name	text	Il nome dell'utente.
role_id	integer	L'ID ruolo del ruolo concesso.
role_name	text	Il nome del ruolo per il ruolo concesso.
admin_option	booleano	Un valore che indica se l'utente può concedere il ruolo ad altri utenti e ruoli.

Query di esempio

Le seguenti query concedono ruoli agli utenti e mostrano l'elenco degli utenti a cui sono espressamente concessi ruoli.

```
GRANT ROLE role1 TO reguser;
GRANT ROLE role2 TO reguser;
GRANT ROLE role1 TO superuser;
GRANT ROLE role2 TO superuser;

SELECT user_name,role_name,admin_option FROM svv_user_grants;

 user_name | role_name | admin_option
-----------+-----------+--------------
 superuser |  role1    | False
 reguser   |  role1    | False
 superuser |  role2    | False
  reguser  |  role2    | False
```

SVV_USER_INFO

Puoi recuperare i dati sugli utenti del database Amazon Redshift con la vista SVV_USER_INFO.

SVV_USER_INFO è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_name	text	Il nome utente del ruolo.
user_id	integer	l'ID utente dell'utente.
createdb	booleano	Un valore che indica se l'utente dispone delle autorizzazioni per creare i database.
superuser	booleano	Un valore che indica se l'utente è un utente con privilegi avanzati.
catalog_update	booleano	Un valore che indica se l'utente può aggiornare i cataloghi di sistema.
connection_limit	text	Il numero di connessioni che l'utente può aprire.
syslog_access	text	Un valore che indica se l'utente dispone dell'acceso ai log di sistema. I due possibili valori sono `RESTRICTED` e `UNRESTRICTED`. `RESTRICTED` indica che gli utenti che non sono utenti con privilegi avanzati possono visualizzare i propri record. `UNRESTRICTED` indica che gli utenti che non sono utenti con privilegi avanzati possono visualizzare tutti i record nelle viste e nelle tabelle di sistema di cui possiedono i privilegi `SELECT`.
last_ddl_timestamp	timestamp	Il timestamp per l'ultima istruzione di creazione data definition language (DDL) eseguita dall'utente.

Nome colonna	Tipo di dati	Descrizione
session_timeout	integer	Il tempo massimo in secondi in cui una sessione rimane inattiva o inattiva prima del timeout. 0 indica che non è impostato alcun timeout. Per informazioni sull'impostazione del timeout inattivo del cluster, consulta Quote e limiti in Amazon Redshift nella Guida alla gestione di Amazon Redshift.
external_user_id	text	L'identificatore univoco dell'utente nel provider di identità di terze parti.

Query di esempio

Il comando seguente recupera le informazioni sull'utente da SVV_USER_INFO.

```
SELECT * FROM SVV_USER_INFO;
```

SVV_VACUUM_PROGRESS

Questa visualizzazione restituisce una stima del tempo che verrà impiegato per completare un'operazione di vacuum attualmente in corso.

SVV_VACUUM_PROGRESS è visibile solo per gli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_VACUUM_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Per informazioni su SVVV_VACUM_SUMMART, consultare SVV_VACUUM_SUMMARY.

Per informazioni su SVL_VACUUM_PERCENTION, consulta SVL_VACUUM_PERCENTAGE.

> **Note**
>
> Questa vista è disponibile solo per le query sui cluster con provisioning.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
table_name	text	Nome della tabella attualmente in fase di vacuum o della tabella che è stata sottoposta a vacuum per ultima se non è in corso alcuna operazione.
status	text	Descrizione dell'attività attualmente svolta come parte dell'operazione di vacuum: • Inizializzazione • Ordina • Unione • Eliminazione • Select • Non riuscito • Completa • Saltato • Creazione dell'ordine INTERLEAVED SORTKEY
time_remaining_estimate	text	Tempo stimato rimanente per il completamento dell'attuale operazione di vacuum, in minuti e secondi: **`5m 10s`**, per esempio. Non viene restituito alcun tempo stimato fino a quando il vacuum non completa la sua prima operazione di ordinamento. Se non è in corso alcun vacuum, l'ultimo vacuum eseguito viene visualizzato come **`Completed`** nella colonna STATUS e in una colonna TIME_REMAINING_ESTIMATE vuota. La stima di solito diventa più accurata con l'avanzare del vacuum.

Query di esempio

Le query seguenti, eseguite a distanza di pochi minuti, mostrano che una tabella grande di nome SALESNEW è in fase di vacuum.

```
select * from svv_vacuum_progress;

table_name    |            status             | time_remaining_estimate
--------------+-------------------------------+-------------------------
salesnew      |  Vacuum: initialize salesnew  |
(1 row)
...
select * from svv_vacuum_progress;

table_name   |         status         | time_remaining_estimate
-------------+------------------------+-------------------------
salesnew     |  Vacuum salesnew sort  | 33m 21s
(1 row)
```

La query seguente mostra che attualmente non è in corso alcuna operazione di vacuum. L'ultima tabella sottoposta a vacuum è stata la tabella SALES.

```
select * from svv_vacuum_progress;

table_name   |  status  | time_remaining_estimate
-------------+----------+-------------------------
  sales      | Complete |
(1 row)
```

SVV_VACUUM_SUMMARY

La visualizzazione SVV_VACUUM_SUMMARY si unisce alle tabelle STL_VACUUM, STL_QUERY e STV_TBL_PERM per riepilogare le informazioni sulle operazioni di vacuum registrate dal sistema. La visualizzazione restituisce una riga per tabella per transazione di vacuum. La visualizzazione registra il tempo trascorso dell'operazione, il numero di partizioni di ordinamento create, il numero di incrementi di unione richiesti e i delta in righe e blocchi prima e dopo l'esecuzione dell'operazione.

SVV_VACUUM_SUMMARY è visibile solo per gli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_VACUUM_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Per informazioni su SVVV_VACUUM_PROGRESS, consultare SVV_VACUUM_PROGRESS.

Per informazioni su SVL_VACUUM_PERCENTION, consulta SVL_VACUUM_PERCENTAGE.

> **Note**
>
> Questa vista è disponibile solo per le query sui cluster con provisioning.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
table_name	text	Nome della tabella sottoposta a vacuum.
xid	bigint	ID di transazione dell'operazione VACUUM.
sort_partitions	bigint	Numero di partizioni ordinate create durante la fase di ordinamento dell'operazione vacuum.
merge_increments	bigint	Numero di incrementi di unione richiesti per completare la fase di unione dell'operazione vacuum.
elapsed_time	bigint	Tempo di esecuzione trascorso dell'operazione vacuum (in microsecondi).
row_delta	bigint	Differenza nel numero totale di righe della tabella prima e dopo il vacuum.
sortedrow_delta	bigint	Differenza nel numero di righe della tabella ordinate prima e dopo il vacuum.
block_delta	integer	Differenza nel conteggio dei blocchi per la tabella prima e dopo il vacuum.

Nome colonna	Tipo di dati	Descrizione
max_merge_partitions	integer	Questa colonna viene utilizzata per l'analisi delle prestazioni e rappresenta il numero massimo di partizioni che il vacuum può elaborare per la tabella per ogni iterazione di fase di unione. (Il vacuum ordina la regione non ordinata in una o più partizioni ordinate. A seconda del numero di colonne nella tabella e della configurazione attuale di Amazon Redshift, la fase di unione può elaborare un numero massimo di partizioni in una singola interazione di unione. La fase di unione continuerà a funzionare se il numero di partizioni ordinate supera il numero massimo di partizioni di unione, ma saranno necessarie altre iterazioni di unione).

Query di esempio

La query seguente restituisce le statistiche per le operazioni di vacuum su tre diverse tabelle. La tabella SALES è stata sottoposta a vacuum due volte.

```
select table_name, xid, sort_partitions as parts, merge_increments as merges,
elapsed_time, row_delta, sortedrow_delta as sorted_delta, block_delta
from svv_vacuum_summary
order by xid;

table_  | xid  |parts|merges| elapsed_ | row_    | sorted_ | block_
name    |      |     |      | time     | delta   | delta   | delta
--------+------+-----+------+----------+---------+---------+--------
users   | 2985 |   1 |    1 | 61919653 |       0 |   49990 |      20
category| 3982 |   1 |    1 | 24136484 |       0 |      11 |       0
sales   | 3992 |   2 |    1 | 71736163 |       0 | 1207192 |      32
sales   | 4000 |   1 |    1 | 15363010 | -851648 | -851648 |    -140
(4 rows)
```

Viste di monitoraggio SYS

Viste di monitoraggio sono viste di sistema in Amazon Redshift utilizzate per monitorare l'utilizzo delle risorse di query e carichi di lavoro dei cluster con provisioning e dei gruppi di lavoro serverless. Queste viste sono situate nello schema `pg_catalog`. Per visualizzare le informazioni fornite da queste visualizzazioni, eseguire istruzioni SQL SELECT.

Salvo diversa indicazione, queste visualizzazioni sono disponibili per i cluster Amazon Redshift e i gruppi di lavoro Amazon Redshift Serverless.

SYS_SERVERLESS_USAGE raccoglie dati sull'utilizzo solo per Amazon Redshift Serverless.

Argomenti

- SYS_PROCEDURE_CALL
- SYS_PROCEDURE_MESSAGES
- SYS_QUERY_DETAIL
- SYS_QUERY_HISTORY
- SYS_QUERY_TEXT
- SYS_RESTORE_LOG
- SYS_RESTORE_STATE
- SYS_SCHEMA_QUOTA_VIOLATIONS
- SYS_SERVERLESS_USAGE
- SYS_SESSION_HISTORY
- SYS_SPATIAL_SIMPLIFY
- SYS_STREAM_SCAN_ERRORS
- SYS_STREAM_SCAN_STATES
- SYS_TRANSACTION_HISTORY
- SYS_UDF_LOG
- SYS_UNLOAD_DETAIL
- SYS_UNLOAD_HISTORY
- SYS_USERLOG
- SYS_VACUUM_HISTORY

SYS_ANALYZE_COMPRESSION_HISTORY

Registra i dettagli per le operazioni di analisi della compressione durante i comandi COPY o ANALYZE COMPRESSION.

SYS_ANALYZE_COMPRESSION_HISTORY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'ID dell'utente che ha generato la voce.
start_time	timestamp	L'ora di inizio dell'operazione di analisi della compressione.
transaction_id	bigint	L'ID transazione dell'operazione di analisi della compressione.
table_id	integer	L'ID della tabella che è stata analizzata.
table_name	character(128)	Il nome della tabella che è stata analizzata.
column_position	integer	L'indice della colonna nella tabella che è stata analizzata per determinare la codifica della compressione.
old_encoding	character(15)	Il tipo di codifica prima dell'analisi della compressione.
new_encoding	character(15)	Il tipo di codifica dopo l'analisi della compressione.
mode	character(14)	I valori possibili sono: PRESET Specifica che `new_encoding` è determinato dal comando Amazon Redshift COPY sulla base del tipo di dati della colonna. Nessun dato viene campionato. ATTIVATO Specifica che `new_encoding` è determinato dal comando Amazon Redshift COPY sulla base di un'analisi dei dati di esempio.

Nome colonna	Tipo di dati	Descrizione
		ANALYZE ONLY Specifica che `new_encoding` è determinato dal comando Amazon Redshift ANALYZE COMPRESSION sulla base di un'analisi dei dati di esempio. Tuttavia, il tipo di codifica della colonna analizzata non viene modificato.

Query di esempio

Nell'esempio seguente vengono esaminati i dettagli dell'analisi della compressione sulla tabella `lineitem` dall'ultimo comando COPY eseguito nella stessa sessione.

```
select transaction_id, table_id, btrim(table_name) as table_name, column_position,
 old_encoding, new_encoding, mode
from sys_analyze_compression_history
where transaction_id = (select transaction_id from sys_query_history where query_id =
 pg_last_copy_id()) order by column_position;

 transaction_id  |  table_id   | table_name | column_position |  old_encoding   |
 new_encoding   |      mode
-----------------+-------------+------------+-----------------+-----------------
+-----------------+-------------
      8196       |   248126    | lineitem   |        0        | mostly32        |
 mostly32        | ON
      8196       |   248126    | lineitem   |        1        | mostly32        | lzo
           | ON
      8196       |   248126    | lineitem   |        2        | lzo             |
 delta32k        | ON
      8196       |   248126    | lineitem   |        3        | delta           | delta
           | ON
      8196       |   248126    | lineitem   |        4        | bytedict        |
 bytedict        | ON
      8196       |   248126    | lineitem   |        5        | mostly32        |
 mostly32        | ON
      8196       |   248126    | lineitem   |        6        | delta           | delta
           | ON
      8196       |   248126    | lineitem   |        7        | delta           | delta
           | ON
```

```
      8196      |   248126    | lineitem   |        8        | lzo             | zstd
           | ON
      8196      |   248126    | lineitem   |        9        | runlength       | zstd
           | ON
      8196      |   248126    | lineitem   |       10        | delta           | lzo
           | ON
      8196      |   248126    | lineitem   |       11        | delta           | delta
           | ON
      8196      |   248126    | lineitem   |       12        | delta           | delta
           | ON
      8196      |   248126    | lineitem   |       13        | bytedict        | zstd
           | ON
      8196      |   248126    | lineitem   |       14        | bytedict        | zstd
           | ON
      8196      |   248126    | lineitem   |       15        | text255         | zstd
           | ON
(16 rows)
```

SYS_ANALYZE_HISTORY

Registra i dettagli delle operazioni ANALYZE.

SYS_ANALYZE_HISTORY è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'ID dell'utente che ha generato la voce.
transaction_id	Long	L'ID transazione.
query_id	Long	L'identificatore della query in SYS_QUERY_HISTORY.
database_name	char(30)	Nome del database.
table_name	char(30)	Nome della tabella.

Nome colonna	Tipo di dati	Descrizione
table_id	integer	L'ID della tabella.
is_automatic	char(1)	Il valore è true (t) se l'operazione includeva per impostazione predefinita un'operazione ANALYZE di Amazon Redshift. Il valore è false (f) se il comando ANALYZE è stato eseguito in modo esplicito.
status	char(15)	Risultato del comando di analisi. I valori possibili sono Full, Skipped e PredicateColumn.
start_time	timestamp	L'ora in UTC in cui è iniziata l'esecuzione dell'operazione ANALYZE.
end_time	timestamp	L'ora in UTC in cui è terminata l'esecuzione dell'operazione ANALYZE.
righe	double	Il numero totale di righe nella tabella
modified_rows	double	Il numero complessivo di righe modificate dopo l'ultima operazione ANALYZE.
analyze_threshold_percent	integer	Il valore del parametro analyze_threshold_percent.
last_analyze_time	timestamp	L'ora in UTC in cui la tabella è stata analizzata in precedenza.

Query di esempio

```
 user_id | transaction_id | database_name | schema_name |      table_name     |
 table_id | is_automatic | Status |         start_time         |          end_time
     | rows | modified_rows | analyze_threshold_percent |  last_analyze_time
---------+----------------+---------------+------------+---------------------
+----------+--------------+--------+----------------------------+----------
```

```
+-----------------+------+--------------+--------------------------
+---------------------
     101 |           8006 |           dev |      public | test_table_562bf8dc
 |   110427 |            f |   Full | 2023-09-21 18:33:08.504646 | 2023-09-21
 18:33:24.296498 |    5 |             5 |                          0 | 2000-01-01
 00:00:00
```

SYS_APPLIED_MASKING_POLICY_LOG

Utilizzate SYS_APPLIED_MASKING_POLICY_LOG per tracciare l'applicazione delle politiche di mascheramento dinamico dei dati sulle query che fanno riferimento a relazioni protette da DDM.

SYS_APPLIED_MASKING_POLICY_LOG è visibile ai seguenti utenti:

- Utenti con privilegi avanzati
- Utenti con il ruolo `sys:operator`
- Utenti con l'autorizzazione ACCESS SYSTEM TABLE

Gli utenti con privilegi normali vedranno 0 righe.

Nota che SYS_APPLIED_MASKING_POLICY_LOG non è visibile agli utenti con il ruolo. `sys:secadmin`

Per ulteriori informazioni sul mascheramento dinamico dei dati, vai Mascheramento dinamico dei dati a.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
nome_policy	text	Nome della policy di mascheramento.
user_id	text	L'ID dell'utente che ha eseguito la query.
record_time	timestamp	L'ora in cui è stata registrata la voce relativa alla visualizzazione del sistema.

Nome colonna	Tipo di dati	Descrizione
session_id	int	L'ID processo.
transaction_id	Long	L'ID transazione.
query_id	int	L'ID di query.
database_name	text	Il nome del database su cui è stata eseguita la query.
relation_name	text	Il nome della tabella a cui viene applicata la politica di mascheramento.
schema_name	text	Il nome dello schema in cui si trova la tabella.
attachment_id	Long	L'ID della politica di mascheramento allegata.
tipo di relazione	text	Il tipo di relazione a cui viene applicata la politica di mascheramento. I valori possibili sono `TABLE`, `VIEW`, `LATE BINDING VIEW` e `MATERIALIZED VIEW` .

Query di esempio

L'esempio seguente mostra che la politica di `mask_credit_card_full` mascheramento è allegata alla `credit_db.public.credit_cards` tabella.

```
select policy_name, database_name, relation_name, schema_name, relation_kind
from sys_applied_masking_policy_log;

policy_name           | database_name | relation_name | schema_name | relation_kind
----------------------+---------------+---------------+-------------+--------------
mask_credit_card_full | credit_db     | credit_cards  | public      | table

(1 row)
```

SYS_AUTO_TABLE_OPTIMIZATION

Registra le azioni automatiche intraprese da Amazon Redshift nelle tabelle definite per l'ottimizzazione automatica.

SYS_AUTO_TABLE_OPTIMIZATION è visibile solo ai superutenti. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
transaction_id	Long	L'identificativo della transazione.
session_id	int	L'identificatore di sessione del processo che ha eseguito il comando alter.
table_id	int	L'identificatore della tabella.
alter_table_type	character (32)	Il tipo di suggerimento. I valori possibili sono `distkey`, `sortkey` e `encode`.
status	character (128)	Lo stato di completamento del suggerimento. I valori possibili sono `Start`, `Complete`, `Skipped`, `Abort`, `Checkpoint` e `Failed`.
event_time	timestamp	Il timestamp della colonna di stato.
alter_from	character (200)	Lo stile di distribuzione precedente e le chiavi di ordinamento della tabella prima di applicare il suggerimento. Il valore viene troncato in incrementi da 200 caratteri.
alter_a	character (200)	Lo stile di distribuzione corrente e le chiavi di ordinamento della tabella dopo l'applicazione del consiglio. Il valore viene troncato in incrementi da 200 caratteri.

Query di esempio

Nell'esempio seguente, le righe del risultato mostrano le azioni intraprese da Amazon Redshift.

```
SELECT table_id, alter_table_type, status, event_time, alter_from
FROM SYS_AUTO_TABLE_OPTIMIZATION;

 table_id |  alter_table_type   |                        status
 |         event_time          |  alter_from
----------+---------------------+-----------------------------------------------------
+-----------------------------+-----------------
   118082 | sortkey             | Start
 | 2020-08-22 19:42:20.727049  |
   118078 | sortkey             | Start
 | 2020-08-22 19:43:54.728819  |
   118082 | sortkey             | Start
 | 2020-08-22 19:42:52.690264  |
   118072 | sortkey             | Start
 | 2020-08-22 19:44:14.793572  |
   118082 | sortkey             | Failed
 | 2020-08-22 19:42:20.728917  |
   118078 | sortkey             | Complete
 | 2020-08-22 19:43:54.792705  |  SORTKEY: None;
   118086 | sortkey             | Complete
 | 2020-08-22 19:42:00.72635   |  SORTKEY: None;
   118082 | sortkey             | Complete
 | 2020-08-22 19:43:34.728144  |  SORTKEY: None;
   118072 | sortkey             | Skipped:Retry exceeds the maximum limit for a table.
 | 2020-08-22 19:44:46.706155  |
   118086 | sortkey             | Start
 | 2020-08-22 19:42:00.685255  |
   118082 | sortkey             | Start
 | 2020-08-22 19:43:34.69531   |
   118072 | sortkey             | Start
 | 2020-08-22 19:44:46.703331  |
   118082 | sortkey             | Checkpoint: progress 14.755079%
 | 2020-08-22 19:42:52.692828  |
   118072 | sortkey             | Failed
 | 2020-08-22 19:44:14.796071  |
   116723 | sortkey             | Abort:This table is not AUTO.
 | 2020-10-28 05:12:58.479233  |
   110203 | distkey             | Abort:This table is not AUTO.
 | 2020-10-28 05:45:54.67259   |
```

SYS_CONNECTION_LOG

Registra i tentativi di autenticazione, insieme alle connessioni.

SYS_CONNECTION_LOG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
evento	character(50)	Evento di connessione o autenticazione.
record_time	timestamp	Ora in cui l'evento si è verificato.
remote_host	character(45)	Nome o indirizzo IP dell'host remoto.
remote_port	character(32)	Numero di porta per l'host remoto.
session_id	integer	ID di processo associato all'istruzione.
database_name	character(50)	Nome del database.
user_name	character(50)	Username.
auth_method	character(32)	Metodo di autenticazione.
durata	integer	Durata di connessione in microsecondi.
ssl_version	character(50)	Versione Secure Sockets Layer (SSL).
ssl_cipher	character(128)	Crittografia SSL.
mtu	integer	Unità di trasmissione massima (MTU).
ssl_compression	character(64)	Tipo di compressione SSL.

Nome colonna	Tipo di dati	Descrizione
ssl_expansion	character(64)	Tipo di espansione SSL.
iam_auth_guid	character(36)	L'ID di autenticazione IAM per la CloudTrail richiesta.
application_name	character(250)	Il nome iniziale o aggiornato dell'applicazione per una sessione.
driver_version	character(64)	La versione del driver ODBC o JDBC che si connette al tuo cluster Amazon Redshift dai tuoi strumenti client SQL di terze parti.
os_version	character(64)	La versione del sistema operativo presente sul computer client che si connette al cluster Amazon Redshift.
plugin_name	character(32)	Il nome del plug-in utilizzato per connettersi al cluster Amazon Redshift.
protocol_version	integer	La versione del protocollo interno utilizzato dal driver Amazon Redshift per stabilire la connessione con il server. Le versioni del protocollo sono negoziate tra driver e server. La versione descrive le funzionalità disponibili. I valori validi includono: • 0 (BASE_SERVER_PROTOCOL_VERSION) • 1 (EXTENDED_RESULT_METADATA_SERVER_PROTOCOL_VERSION): per salvare un'andata e ritorno per query, il server invia ulteriori informazioni sui metadati del set di risultati. • 2 (BINARY_PROTOCOL_VERSION): a seconda del tipo di dati del set di risultati, il server invia dati in formato binario. • 3 (EXTENDED2_RESULT_METADATA_SERVER_PROTOCOL_VERSION): il server invia informazioni di distinzione tra maiuscole e minuscole (regole di confronto) di una colonna.

Nome colonna	Tipo di dati	Descrizione
global_session_id	character(36)	L'identificatore univoco globale per la sessione attuale. L'ID di sessione persiste con il riavvio dell'errore del nodo.

Query di esempio

Per visualizzare i dettagli per le connessioni aperte, esegui la seguente query.

```
select record_time, user_name, database_name, remote_host, remote_port
from sys_connection_log
where event = 'initiating session'
and session_id not in
(select session_id from sys_connection_log
where event = 'disconnecting session')
order by 1 desc;

record_time         | user_name   | database_name   | remote_host   | remote_port

--------------------+-------------+-----------------+---------------
+---------------------------------
2014-11-06 20:30:06 | rdsdb       | dev             | [local]       |

2014-11-06 20:29:37 | test001     | test            | 10.49.42.138  | 11111

2014-11-05 20:30:29 | rdsdb       | dev             | 10.49.42.138  | 33333

2014-11-05 20:28:35 | rdsdb       | dev             | [local]       |
(4 rows)
```

L'esempio seguente riflette un tentativo di autenticazione non riuscito e una connessione e una disconnessione andate a buon fine.

```
select event, record_time, remote_host, user_name
from sys_connection_log order by record_time;

            event      |         record_time        |  remote_host  | user_name

-----------------------+----------------------------+---------------+---------
```

```
authentication failure | 2012-10-25 14:41:56.96391  | 10.49.42.138  | john

authenticated          | 2012-10-25 14:42:10.87613  | 10.49.42.138  | john

initiating session     | 2012-10-25 14:42:10.87638  | 10.49.42.138  | john

disconnecting session  | 2012-10-25 14:42:19.95992  | 10.49.42.138  | john

(4 rows)
```

L'esempio seguente mostra la versione del driver ODBC, il sistema operativo sul computer client e il plug-in utilizzato per connettersi al cluster Amazon Redshift. In questo esempio, il plug-in utilizzato è per l'autenticazione standard del driver ODBC utilizzando un nome di accesso e una password.

```
select driver_version, os_version, plugin_name from sys_connection_log;

driver_version                          |  os_version                       |
 plugin_name
-----------------------------------------+-----------------------------------
+--------------------
Amazon Redshift ODBC Driver 1.4.15.0001 | Darwin 18.7.0 x86_64              | none
Amazon Redshift ODBC Driver 1.4.15.0001 | Linux 4.15.0-101-generic x86_64   | none
```

Nell'esempio seguente viene illustrata la versione del sistema operativo sul computer client, la versione del driver e la versione del protocollo.

```
select os_version, driver_version, protocol_version from sys_connection_log;

os_version                     |  driver_version              | protocol_version
-------------------------------+------------------------------+--------------------
Linux 4.15.0-101-generic x86_64 | Redshift JDBC Driver 2.0.0.0 | 2
Linux 4.15.0-101-generic x86_64 | Redshift JDBC Driver 2.0.0.0 | 2
Linux 4.15.0-101-generic x86_64 | Redshift JDBC Driver 2.0.0.0 | 2
```

SYS_COPY_JOB (anteprima)

Questa è la documentazione non definitiva per autocopy (SQL COPY JOB), disponibile nella versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consiglia mo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. L'anteprima pubblica terminerà il 31 luglio 2024. I cluster di anteprima verranno rimossi automatic

amente due settimane dopo il termine dell'anteprima. Per i termini e le condizioni dell'anteprima, consulta la sezione relativa a beta e anteprime nei termini del servizio AWS.

Utilizza SYS_COPY_JOB per visualizzare i dettagli dei comandi COPY JOB.

Questa vista contiene i comandi COPY JOB che sono stati creati.

SYS_COPY_JOB è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
job_id	bigint	Identificatore del processo di copia.
job_name	character(128)	Nome del processo di copia.
iam_role	character(128)	Ruolo IAM specificato nell'istruzione COPY.
job_text	character(256)	Parametri dell'istruzione COPY.
is_auto	integer	Indica se il processo COPY JOB viene eseguito automaticamente da Amazon Redshift. Il valore 1 indica True, mentre il valore 0 indica False.
on_error_suspend	integer	Queste informazioni sono solo per uso interno.

SYS_COPY_REPLACEMENTS

Mostra un log che registra il momento in cui dei caratteri UTF-8 non validi sono stati sostituiti dal comando COPY con l'opzione ACCEPTINVCHARS. Una voce di log viene aggiunta a SYS_COPY_REPLACEMENTS per ognuna delle prime 100 righe su ogni sezione di nodo che ha richiesto almeno una sostituzione.

È possibile utilizzare questa visualizzazione per visualizzare informazioni sui gruppi di lavoro serverless e sui cluster predisposti.

SYS_COPY_REPLACEMENTS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	ID dell'utente che ha generato la query.
query_id	bigint	L'ID di query. La colonna utilizzata per unire altre tabelle e visualizzazioni del sistema.
table_id	integer	L'ID della tabella.
file_name	character (256)	Percorso completo verso il file di input per il comando COPY.
column_name	character (127)	Primo campo che conteneva un carattere UTF-8 non valido.
line_number	bigint	Il numero di riga nel file di dati di input che contiene un carattere UTF-8 non valido. -1 indica che il numero di riga non è disponibile, ad esempio quando si copia da un file di dati a colonne.
raw_line	character (1024)	Dati di caricamento non elaborati che contenevano un carattere UTF-8 non valido.

Query di esempio

L'esempio seguente restituisce le sostituzioni per l'operazione COPY più recente.

```
select query_idp, table_id, file_name, line_number, colname
from sys_copy_replacements
where query = pg_last_copy_id();

 query_id | table_id |   file_name                          | line_number | column_name
 ---------+----------+--------------------------------------+-------------+--------
    96    |    26    | s3://mybucket/allusers_pipe.txt   |         123 | city
    96    |    26    | s3://mybucket/allusers_pipe.txt   |         456 | city
    96    |    26    | s3://mybucket/allusers_pipe.txt   |         789 | city
    96    |    26    | s3://mybucket/allusers_pipe.txt   |         012 | city
    96    |    26    | s3://mybucket/allusers_pipe.txt   |         119 | city
...
```

SYS_DATASHARE_CHANGE_LOG

Registra la vista consolidata per tenere traccia delle modifiche apportate alle unità di condivisione dati nei cluster di produttori e consumer.

SYS_DATASHARE_CHANGE_LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'ID dell'utente che esegue l'operazione.
user_name	varchar(128)	Il nome dell'utente che esegue l'operazione.
session_id	integer	L'ID della sessione.
transaction_id	bigint	L'ID della transazione.

Nome colonna	Tipo di dati	Descrizione
share_id	integer	L'ID dell'unità di condivisione dati è interessato.
share_name	varchar(128)	Il nome dell'unità di condivisione dati.
source_database_id	integer	L'ID del database a cui appartiene l'unità di condivisione dati.
source_database_name	varchar(128)	Il nome del database a cui appartiene l'unità di condivisione dati.
consumer_database_id	integer	ID del database importato dall'unità di condivisione dati.
consumer_database_name	varchar(128)	Il nome del database importato dall'unità di condivisione dati.
arn	varchar(192)	L'ARN della risorsa che supporta il database importato.
record_time	timestamp	Il timestamp dell'operazione.
action	varchar(128)	L'operazione in esecuzione. I valori possibili sono CREATE DATASHARE, DROP DATASHARE, GRANT ALTER, REVOKE ALTER, GRANT SHARE, REVOKE SHARE, ALTER ADD, ALTER REMOVE, ALTER SET, GRANT USAGE, REVOKE USAGE, CREATE DATABASE, GRANT o REVOKE USAGE su un database condiviso, DROP SHARED DATABASE, ALTER SHARED DATABASE.
status	integer	Lo stato dell'operazione. I valori possibili sono SUCCESS e ERROR-ERROR CODE

Nome colonna	Tipo di dati	Descrizione
share_object_type	varchar(64)	Il tipo di oggetto di database aggiunto o rimosso dall'unità di condivisione dati. I valori possibili sono schemi, tabelle, colonne, funzioni e viste. Questo è un campo per il cluster di produttori.
share_object_id	integer	L'ID dell'oggetto di database aggiunto o rimosso dall'unità di condivisione dati. Questo è un campo per il cluster di produttori.
share_object_name	varchar(128)	Il nome dell'oggetto di database aggiunto o rimosso dall'unità di condivisione dati. Questo è un campo per il cluster di produttori.
target_user_type	varchar(16)	Il tipo di utenti o gruppi a cui è stato concesso un privilegio. Questo è un campo per il cluster di produttori e consumer.
target_user_id	integer	L'ID di utenti o gruppi a cui è stato concesso un privilegio. Questo è un campo per il cluster di produttori e consumer.
target_user_name	varchar(128)	Il nome di utenti o gruppi a cui è stato concesso un privilegio. Questo è un campo per il cluster di produttori e consumer.
consumer_account	varchar(16)	L'ID account del consumer di dati. Questo è un campo per il cluster di produttori.
consumer_namespace	varchar(64)	Lo spazio dei nomi dell'account del consumer di dati. Questo è un campo per il cluster di produttori.
producer_account	varchar(16)	L'ID account dell'account del produttore a cui appartiene l'unità di condivisione dati. Questo è un campo per il cluster di consumer.
producer_namespace	varchar(64)	Lo spazio dei nomi dell'account di produzione a cui appartiene l'unità di condivisione dati. Questo è un campo per il cluster di consumer.
attribute_name	varchar(64)	Il nome di un attributo dell'unità di condivisione dati o del database condiviso.
attribute_value	varchar(128)	Il valore di un attributo dell'unità di condivisione dati o del database condiviso.

Nome colonna	Tipo di dati	Descrizione
message	varchar(512)	Il messaggio di errore visualizzato quando un'operazione non riesce.

Query di esempio

L'esempio seguente mostra una vista SYS_DATASHARE_CHANGE_LOG.

```
SELECT DISTINCT action
FROM sys_datashare_change_log
WHERE share_object_name LIKE 'tickit%';

         action
 ----------------------
  "ALTER DATASHARE ADD"
```

SYS_DATASHARE_CROSS_REGION_USAGE

Utilizza la vista SYS_DATASHARE_CROSS_REGION_USAGE per ottenere un riepilogo dell'utilizzo dei dati trasferiti tra regioni causato da query dell'unità di condivisione dati tra regioni. SYS_DATASHARE_CROSS_REGION_USAGE aggrega i dettagli a livello di segmento.

SYS_DATASHARE_CROSS_REGION_USAGE è visibile solo per gli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query_id	integer	L'ID della query. Utilizza questo valore per unire altre tabelle e viste di sistema.
child_query_sequence	integer	La sequenza della query utente riscritta, che inizia con 1.

Nome colonna	Tipo di dati	Descrizione
segment_id	bigint	Il numero del segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi.
start_time	time	L'ora in UTC in cui è iniziato il trasferimento dei dati.
end_time	time	L'ora in UTC in cui è terminato il trasferimento dei dati.
transferred_data	bigint	Il numero di byte di dati trasferiti da una regione producer a una regione consumer.
source_region	char(25)	La regione producer da cui la query ha trasferito i dati.

Query di esempio

L'esempio seguente mostra una vista SYS_DATASHARE_CROSS_REGION_USAGE.

```
SELECT query, segment, transferred_data, source_region
from sys_datashare_cross_region_usage
where query = pg_last_query_id()
order by query,segment;

  query | segment | transferred_data | source_region
--------+---------+------------------+---------------
 200048 |       2 |          4194304 |    us-west-1
 200048 |       2 |          4194304 |    us-east-2
```

SYS_DATASHARE_USAGE_CONSUMER

Registra l'attività e l'utilizzo delle unità di condivisione dati. Questa visualizzazione è rilevante solo per il cluster di consumer.

SYS_DATASHARE_USAGE_CONSUMER è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'ID dell'utente che invia la richiesta.
session_id	integer	L'ID del processo leader che esegue la query.
transaction_id	bigint	Il contesto della transazione corrente.
request_id	varchar(50)	L'ID univoco della chiamata API richiesta.
request_type	varchar(25)	Il tipo di richiesta presentata al cluster di produttori.
transaction_uid	varchar(50)	L'ID univoco della transazione.
record_time	timestamp	L'ora in cui viene registrata l'operazione.
status	integer	Lo stato della chiamata API richiesta.
error_message	varchar(512)	Il messaggio per un errore.

Query di esempio

L'esempio seguente mostra la vista SYS_DATASHARE_USAGE_CONSUMER.

```
SELECT request_type, status, trim(error) AS error
FROM sys_datashare_usage_consumer

  request_type  | status | error_message
----------------+--------+---------------
 "GET RELATION" |   0    |
```

SYS_DATASHARE_USAGE_PRODUCER

Registra l'attività e l'utilizzo delle unità di condivisione dati. Questa visualizzazione è rilevante solo per il cluster di consumer.

SYS_DATASHARE_USAGE_PRODUCER è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
share_id	integer	L'ID oggetto (OID) dell'unità di condivisione dati.
share_name	varchar(128)	Il nome dell'unità di condivisione dati.
request_id	varchar(50)	L'ID univoco della chiamata API richiesta.
request_type	varchar(25)	Il tipo di richiesta presentata al cluster di produttori.
object_type	varchar(64)	Il tipo di oggetto condiviso dall'unità di condivisione dati. I valori possibili sono schemi, tabelle, colonne, funzioni e viste.
object_oid	integer	L'ID oggetto condiviso dall'unità di condivisione dati.
object_name	varchar(128)	Il nome dell'oggetto condiviso dall'unità di condivisione dati.
consumer_account	varchar(16)	L'account dell'account consumer con cui è condivisa l'unità di condivisione dati.
consumer_namespace	varchar(64)	Lo spazio dei nomi dell'account consumer con cui è condivisa l'unità di condivisione dati.

Nome colonna	Tipo di dati	Descrizione
consumer_transaction_uid	varchar(50)	L'ID transazione univoco dell'istruzione nel cluster di consumer.
record_time	timestamp	L'ora in cui viene registrata l'operazione.
status	integer	Lo stato dell'unità di condivisione dati.
error_message	varchar(512)	Il messaggio per un errore.
consumer_region	char(64)	La regione in cui si trova il cluster consumer.

Query di esempio

L'esempio seguente mostra la vista SYS_DATASHARE_USAGE_PRODUCER.

```
SELECT DISTINCT
FROM sys_datashare_usage_producer
WHERE object_name LIKE 'tickit%';

   request_type
 ------------------
   "GET RELATION"
```

SYS_EXTERNAL_QUERY_DETAIL

Utilizzare SYS_EXTERNAL_QUERY_DETAIL per visualizzare i dettagli delle query a livello di segmento. Ogni riga rappresenta un segmento di una determinata query WLM con dettagli quali il numero di righe elaborate, il numero di byte elaborati e le informazioni sulle partizioni delle tabelle esterne in Amazon S3. Ogni riga di questa vista avrà anche una voce corrispondente nella vista SYS_QUERY_DETAIL, tranne che questa vista contiene informazioni più dettagliate relative all'elaborazione delle query esterne.

SYS_EXTERNAL_QUERY_DETAIL è visibile per tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	Identificatore dell'utente che ha inviato la query.
query_id	bigint	L'identificatore della query esterna.
transaction_id	bigint	L'identificativo della transazione.
child_query_sequence	integer	La sequenza della query utente riscritta. Inizia con 0, simile a segment_id.
segment_id	integer	L'identificatore di segmento del segmento di query.
source_type	character(32)	Il tipo di origine dati della query, potrebbe essere S3 per Redshift Spectrum, PG per query federata.
start_time	timestamp	Il momento in cui è iniziata la query.
end_time	timestamp	L'ora in cui è stata completata la query.
durata	bigint	La quantità di tempo (microsecondi) dedicato alla query.

Nome colonna	Tipo di dati	Descrizione
total_partitions	integer	Il numero di partizioni necessarie per una query Amazon S3.
qualified_partitions	integer	Il numero di partizioni sottoposte a scansione di una query Amazon S3.
scanned_files	bigint	Il numero di file Amazon S3 sottoposti a scansione.
returned_rows	bigint	Il numero di righe scansionate per una query Amazon S3 o il numero di righe restituite per una query federata.
returned_bytes	bigint	Il numero di byte scansionati per una query Amazon S3 o il numero di byte restituiti per una query federata.
file_format	text	Il formato di file dei file Amazon S3.
file_location	text	La posizione di Amazon S3 della tabella esterna.
external_query_text	text	Il testo della query a livello di segmento per una query federata.
warning_message	character(4000)	Il messaggio di avviso visualizzato durante l'esecuzi one della query.
table_name	character(136)	Nome della tabella del passaggio in corso.

Nome colonna	Tipo di dati	Descrizione
is_recursive	character(1)	Indica se è presente la scansione ricorsiva delle sottocartelle.
is_nested	character(1)	Indica se è possibile accedere al tipo di dati della colonna nidificata.
s3list_time	bigint	La durata dell'elenco dei file in millisecondi.
get_partition_time	Long	Tempo impiegato per elencare e qualificare le partizioni per un determinato oggetto esterno da Apache Hive. AWS Glue Data Catalog

Query di esempio

La seguente query mostra i dettagli della query esterna.

```
SELECT query_id,
       segment_id,
       start_time,
       end_time,
       total_partitions,
       qualified_partitions,
       scanned_files,
       returned_rows,
       returned_bytes,
       trim(external_query_text) query_text,
       trim(file_location) file_location
FROM sys_external_query_detail
ORDER BY query_id, start_time DESC
LIMIT 2;
```

Output di esempio.

```
 query_id | segment_id |         start_time         |          end_time
   | total_partitions | qualified_partitions | scanned_files | returned_rows |
 returned_bytes | query_text | file_location
----------+------------+----------------------------+----------------------------
+-----------------+---------------------+--------------+--------------
+----------------+------------+--------------
   763251 |          0 | 2022-02-15 22:32:23.312448 | 2022-02-15 22:32:24.036023 |
          3 |                   3 |             3 |         38203 |        2683414 |
          |
   763254 |          0 | 2022-02-15 22:32:40.17103  | 2022-02-15 22:32:40.839313 |
          3 |                   3 |             3 |         38203 |        2683414 |
          |
```

SYS_EXTERNAL_QUERY_ERROR

Scopri come eseguire query sulla vista di sistema SYS_EXTERNAL_QUERY_ERROR per ottenere informazioni sugli errori di scansione Redshift Spectrum. SYS_EXTERNAL_QUERY_ERROR visualizza un esempio di errori registrati. Il valore di default è 10 voci per query.

SYS_EXTERNAL_QUERY_ERROR è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'ID dell'utente che ha generato questa riga.
query_id	bigint	L'ID della query che ha generato questa riga.
file_location	char(256)	La posizione dei dati da interrogare.
rowid	char(2.100)	La posizione dell'errore all'interno del file. Le parti `rowid` sono separate da `:` (due punti); è possibile che altre parti vengano aggiunte in futuro. *`row_offset :row_group :row_id`*

Nome colonna	Tipo di dati	Descrizione
		row_offset rappresenta l'offset (in byte) della riga all'interno del file ed è impostato su -1 per i formati di file non supportati. Una tabella è divisa in row_groups e ogni gruppo include righe con row_id distinti.
column_name	char(127)	Il nome della colonna restituito dalla query.
original_value	char(1024)	Valore originale interrogato.
modified_value	char(1024)	Valore modificato restituito in base all'opzione di configurazione di gestione dei dati specificata nella query.
Trigger	char(128)	Opzione di gestione dati specificata nella query.
action	char(128)	Azione associata all'opzione di gestione dei dati specificata nella query.
action_value	char(128)	Valore del parametro dell'operazione associato all'opzione di gestione dei dati specificata nella query.
error_code	integer	Codice risultato dell'opzione di gestione dati specificata nella query.

Query di esempio

La seguente query restituisce le righe per le quali sono state eseguite operazioni di gestione dei dati.

```
SELECT * FROM sys_external_query_error;
```

La query restituisce risultati simili a quanto esposto di seguito.

```
  user_id   query_id  file_location                                rowid
column_name           original_value            modified_value       trigger
action               action_value                error_code
```

```
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:0
league_name           Barclays Premier League    Barclays Premier Lea UNSPECIFIED
TRUNCATE                                          156
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:0
league_nspi           34595                      32767                UNSPECIFIED
OVERFLOW_VALUE                                    199
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:1
league_nspi           34151                      32767                UNSPECIFIED
OVERFLOW_VALUE                                    199
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:2
league_name           Barclays Premier League    Barclays Premier Lea UNSPECIFIED
TRUNCATE                                          156
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:2
league_nspi           33223                      32767                UNSPECIFIED
OVERFLOW_VALUE                                    199
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:3
league_name           Barclays Premier League    Barclays Premier Lea UNSPECIFIED
TRUNCATE                                          156
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:3
league_nspi           32808                      32767                UNSPECIFIED
OVERFLOW_VALUE                                    199
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:4
league_nspi           32790                      32767                UNSPECIFIED
OVERFLOW_VALUE                                    199
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:5
league_name           Spanish Primera Division   Spanish Primera Divi UNSPECIFIED
TRUNCATE                                          156
    100     1574007   s3://spectrum-uddh/league/spi_global_rankings.0:6
league_name           Spanish Primera Division   Spanish Primera Divi UNSPECIFIED
TRUNCATE                                          156
```

SYS_INTEGRATION_ACTIVITY

SYS_INTEGRATION_ACTIVITY visualizza di dettagli relativi alle esecuzioni completate delle integrazioni.

SYS_INTEGRATION_ACTIVITY è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni sulle integrazioni Zero-ETL, consulta Utilizzo delle integrazioni Zero-ETL nella Guida alla gestione di Amazon Redshift.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
integration_id	character (128)	L'identificatore associato all'integrazione.
target_database	character (128)	Il database in Amazon Redshift che riceve i dati dell'integrazione.
source	character (128)	I dati di origine per l'integrazione. I tipi possibili sono `MySQL` e `PostgreSQL` .
checkpoint_name	character (128)	Il nome del checkpoint che replica le coordinate binlog.
checkpoint_type	character (16)	Il tipo di checkpoint. I valori possibili includono: `snapshot`, `cdc`.
checkpoint_bytes	bigint	Il numero di byte in questo checkpoint.
last_commit_timestamp	timestamp	Il timestamp dell'ultima volta che è stato eseguito il commit in questo checkpoint.
modified_tables	integer	Il numero di tabelle modificate nel checkpoint.
integration_start_time	time	L'ora (UTC) in cui è iniziata l'integrazione per questo checkpoint.
integration_end_time	time	L'ora (UTC) in cui è finita l'integrazione per questo checkpoint.

Query di esempio

Il seguente comando SQL mostra il log delle integrazioni.

```
select * from sys_integration_activity;

          integration_id              | target_database | source |
   checkpoint_name                   | checkpoint_type  | checkpoint_bytes |
```

```
 last_commit_timestamp   | modified_tables |   integration_start_time   |
 integration_end_time
-------------------------------------+----------------+--------
+-----------------------------------------+-----------------+-----------------
+-------------------------+----------------+---------------------------
+---------------------------
 76b15917-afae-4447-b7fd-08e2a5acce7b |   demo1         | MySQL  | checkpoints/
checkpoint_3_241_3_510.json     |        cdc       |        762       | 2023-05-10
 23:00:14.201 |         1       | 2023-05-10 23:00:45.054265 | 2023-05-10
 23:00:46.339826
 76b15917-afae-4447-b7fd-08e2a5acce7b |   demo1         | MySQL  | checkpoints/
checkpoint_3_16329_3_17839.json |        cdc       |       13488      | 2023-05-11
 01:33:57.411 |         2       | 2023-05-11 02:19:09.440121 | 2023-05-11
 02:19:16.090492
 76b15917-afae-4447-b7fd-08e2a5acce7b |   demo1         | MySQL  | checkpoints/
checkpoint_3_5103_3_5532.json   |        cdc       |        1657      | 2023-05-10
 23:13:14.205 |         2       | 2023-05-10 23:13:23.545487 | 2023-05-10
 23:13:25.652144
```

SYS_INTEGRATION_TABLE_STATE_CHANGE

SYS_INTEGRATION_TABLE_STATE_CHANGE visualizza i dettagli sui registri delle modifiche dello stato della tabella per le integrazioni.

Un utente con privilegi avanzati può visualizzare tutte le righe della tabella.

Per ulteriori informazioni, consulta Lavorare con le integrazioni Zero-ETL.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
integration_id	character (128)	L'identificatore associato all'integrazione.
database_name	character (128)	Il nome del database Amazon Redshift.
schema_name	character (128)	Nome dello schema Amazon Redshift.

Nome colonna	Tipo di dati	Descrizione
table_name	character (128)	Nome della tabella.
new_state	character (128)	Lo stato della tabella. I valori possibili sono `Synced`, `ResyncRequired` , `ResyncInitiated` , `Deleted`, `Failed` e `ResyncDeleted` .
table_last_replicated_checkpoint	character (128)	Le coordinate attuali del log sincronizzato.
state_change_reason	character (256)	Il motivo dell'ultima transizione dello stato.
record_time	timestamp	L'ora (UTC) in cui questo record è stato aggiornato.

Query di esempio

Il seguente comando SQL mostra il log delle integrazioni.

```
select * from sys_integration_table_state_change;

            integration_id            | database_name | schema_name | table_name
 | new_state |  table_last_replicated_checkpoint   | state_change_reason |
 record_time
--------------------------------------+---------------+-------------+------------
+-----------+--------------------------------------+---------------------
+----------------------------
 99108e72-1cfd-414f-8cc0-0216acefac77 | perfdb        | sbtest80t3s | sbtest79   |
 Synced    | {"txn_seq":9834,"txn_id":126597515} |                     | 2023-09-20
 19:39:50.087868
 99108e72-1cfd-414f-8cc0-0216acefac77 | perfdb        | sbtest80t3s | sbtest56   |
 Synced    | {"txn_seq":9834,"txn_id":126597515} |                     | 2023-09-20
 19:39:45.54005
 99108e72-1cfd-414f-8cc0-0216acefac77 | perfdb        | sbtest80t3s | sbtest50   |
 Synced    | {"txn_seq":9834,"txn_id":126597515} |                     | 2023-09-20
 19:40:20.362504
 99108e72-1cfd-414f-8cc0-0216acefac77 | perfdb        | sbtest80t3s | sbtest18   |
 Synced    | {"txn_seq":9834,"txn_id":126597515} |                     | 2023-09-20
 19:40:32.544084
```

```
99108e72-1cfd-414f-8cc0-0216acefac77 | perfdb        | sbtest40t3s | sbtest23   |
Synced    | {"txn_seq":9834,"txn_id":126597515} |                    | 2023-09-20
15:49:05.186209
```

SYS_LOAD_DETAIL

Restituisce informazioni per tracciare o risolvere i problemi di caricamento dati.

Questa vista registra il progresso di tutti i file di dati mentre vengono caricati in una tabella del database.

Questa vista è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	ID dell'utente che ha generato la voce.
query_id	integer	ID query.
file_name	character(256)	Il nome del file da caricare.
bytes_scanned	integer	Il numero di byte sottoposti a scansione da Amazon S3.
lines_scanned	integer	Numero di righe analizzate dal file di importazione. Questo numero può non corrispondere al numero di righe effettivamente caricate. Ad esempio, il carico può analizzare e tollerare un numero di record danneggiati, basato sull'opzione MAXERROR nel comando COPY.
record_time	timestamp	L'ultima volta in cui la voce è stata aggiornata.
splits_scanned	Numero di suddivisioni di questo file.	Numero di suddivisioni di questo file.

Nome colonna	Tipo di dati	Descrizione
start_time	timestamp	Ora di inizio dell'elaborazione del file.
end_time	timestamp	Ora di fine dell'elaborazione del file.

Query di esempio

L'esempio seguente restituisce i dettagli per l'ultima operazione COPY.

```
select query_id, trim(file_name) as file, record_time
from sys_load_detail
where query_id = pg_last_copy_id();

 query_id |               file                |          record_time
----------+-----------------------------------+----------------------------
 28554    | s3://dw-tickit/category_pipe.txt | 2013-11-01 17:14:52.648486
(1 row)
```

La query seguente contiene voci per un nuovo carico delle tabelle nel database TICKIT:

```
select query_id, trim(file_name), record_time
from sys_load_detail
where file_name like '%tickit%' order by query_id;

 query_id |           btrim           |          record_time
----------+---------------------------+----------------------------
 22475    | tickit/allusers_pipe.txt | 2013-02-08 20:58:23.274186
 22478    | tickit/venue_pipe.txt    | 2013-02-08 20:58:25.070604
 22480    | tickit/category_pipe.txt | 2013-02-08 20:58:27.333472
 22482    | tickit/date2008_pipe.txt | 2013-02-08 20:58:28.608305
 22485    | tickit/allevents_pipe.txt| 2013-02-08 20:58:29.99489
 22487    | tickit/listings_pipe.txt | 2013-02-08 20:58:37.632939
 22593    | tickit/allusers_pipe.txt | 2013-02-08 21:04:08.400491
 22596    | tickit/venue_pipe.txt    | 2013-02-08 21:04:10.056055
 22598    | tickit/category_pipe.txt | 2013-02-08 21:04:11.465049
 22600    | tickit/date2008_pipe.txt | 2013-02-08 21:04:12.461502
 22603    | tickit/allevents_pipe.txt| 2013-02-08 21:04:14.785124
 22605    | tickit/listings_pipe.txt | 2013-02-08 21:04:20.170594
```

```
(12 rows)
```

Il fatto che un record sia scritto nel file di log per questa vista di sistema non significa che il carico abbia eseguito il commit in modo efficiente come parte della transazione contenente. Per verificare i commit di carico, esegui una query sulla vista STL_UTILITYTEXT e cerca il record COMMIT che corrisponde a una transazione COPY. Ad esempio, questa query unisce SYS_LOAD_DETAIL e STL_QUERY sulla base di una sottoquery su STL_UTILITYTEXT:

```
select l.query_id,rtrim(l.file_name),q.xid
from sys_load_detail l, stl_query q
where l.query_id=q.query
and exists
(select xid from stl_utilitytext where xid=q.xid and rtrim("text")='COMMIT');

 query_id |           rtrim           |  xid
----------+---------------------------+-------
 22600    | tickit/date2008_pipe.txt  | 68311
 22480    | tickit/category_pipe.txt  | 68066
  7508    | allusers_pipe.txt         | 23365
  7552    | category_pipe.txt         | 23415
  7576    | allevents_pipe.txt        | 23429
  7516    | venue_pipe.txt            | 23390
  7604    | listings_pipe.txt         | 23445
 22596    | tickit/venue_pipe.txt     | 68309
 22605    | tickit/listings_pipe.txt  | 68316
 22593    | tickit/allusers_pipe.txt  | 68305
 22485    | tickit/allevents_pipe.txt | 68071
  7561    | allevents_pipe.txt        | 23429
  7541    | category_pipe.txt         | 23415
  7558    | date2008_pipe.txt         | 23428
 22478    | tickit/venue_pipe.txt     | 68065
   526    | date2008_pipe.txt         |  2572
  7466    | allusers_pipe.txt         | 23365
 22482    | tickit/date2008_pipe.txt  | 68067
 22598    | tickit/category_pipe.txt  | 68310
 22603    | tickit/allevents_pipe.txt | 68315
 22475    | tickit/allusers_pipe.txt  | 68061
   547    | date2008_pipe.txt         |  2572
 22487    | tickit/listings_pipe.txt  | 68072
  7531    | venue_pipe.txt            | 23390
  7583    | listings_pipe.txt         | 23445
(25 rows)
```

SYS_LOAD_ERROR_DETAIL

Utilizzare SYS_LOAD_ERROR_DETAIL per visualizzare i dettagli degli errori di comando COPY. Ogni riga rappresenta un comando COPY. Contiene comandi COPY in esecuzione e finiti.

SYS_LOAD_ERROR_DETAIL è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'identificatore dell'utente che ha inviato la copia.
query_id	bigint	L'identificatore query della copia.
transaction_id	bigint	L'identificativo della transazione.
session_id	integer	L'identificatore processo del processo che esegue la copia.
database_name	character(64)	Il nome del database al quale l'utente era collegato al momento del rilascio della copia.
table_id	integer	L'identificatore della tabella.
start_time	timestamp	L'ora (UTC) in cui è iniziata la copia.
file_name	character(256)	Il percorso completo verso il file di input da caricare.
line_number	bigint	Il numero di riga nel file di importazione con l'errore.

Nome colonna	Tipo di dati	Descrizione
		Quando si carica un file JSON, il numero di riga dell'ultima riga dell'oggetto JSON con l'errore.
column_name	character(127)	Il campo con l'errore
column_type	character(10)	Il tipo di dati del campo con l'errore.
column_length	character(10)	La lunghezza della colonna, se applicabile. Questo campo è popolato quando il tipo di dati ha una lunghezza limite. Ad esempio, per una colonna con un tipo di dati di "characte r(3)", questa colonna contiene il valore "3 ".
posizione	integer	La posizione dell'errore nel campo.
error_code	integer	Il codice di errore.
error_message	character(512)	La spiegazione dell'errore.

Query di esempio

La seguente query mostra i dettagli dell'errore di caricamento del comando di copia per una query specifica.

```
SELECT query_id,
       table_id,
       start_time,
       trim(file_name) AS file_name,
       trim(column_name) AS column_name,
       trim(column_type) AS column_type,
```

```
       trim(error_message) AS error_message
FROM sys_load_error_detail
WHERE query_id = 762949
ORDER BY start_time
LIMIT 10;
```

Output di esempio.

```
 query_id | table_id |         start_time         |               file_name
      | column_name | column_type |                 error_message
----------+----------+----------------------------
+----------------------------------------+------------+------------
+-----------------------------------------------
   762949 |   137885 | 2022-02-15 22:14:46.759151 | s3://load-test/copyfail/
wrong_format_000 | id          | int4        | Invalid digit, Value 'a', Pos 0, Type:
 Integer
   762949 |   137885 | 2022-02-15 22:14:46.759151 | s3://load-test/copyfail/
wrong_format_001 | id          | int4        | Invalid digit, Value 'a', Pos 0, Type:
 Integer
```

SYS_LOAD_HISTORY

Utilizzare SYS_LOAD_HISTORY per visualizzare i dettagli dei comandi COPY. Ogni riga rappresenta un comando COPY con statistiche accumulate per alcuni campi. Contiene comandi COPY in esecuzione e finiti.

SYS_LOAD_HISTORY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'identificatore dell'utente che ha inviato la copia.
query_id	bigint	L'identificatore query della copia.

Nome colonna	Tipo di dati	Descrizione
transaction_id	bigint	L'identificativo della transazione.
session_id	integer	L'identificatore processo del processo che esegue la copia.
database_name	text	Il nome del database al quale l'utente era collegato al momento del rilascio dell'operazione.
status	text	Lo stato della copia. I valori validi sono `running`, `completed` , `aborted`.
table_name	text	Nome della tabella in cui è stata copiata.
start_time	timestamp	Il momento in cui è iniziata la copia.
end_time	timestamp	Il momento in cui la copia è stata completata.
durata	bigint	Il tempo trascorso (microsecondi) nel comando COPY.
data_source	text	La posizione di Amazon S3 dei file di input da copiare.
file_format	text	Il formato del file di origine. I formati includono csv, txt, json, avro, orc o parquet.
loaded_rows	bigint	Il numero di righe copiate in una tabella.

Nome colonna	Tipo di dati	Descrizione
loaded_bytes	bigint	Il numero di byte copiati in una tabella.
source_file_count	integer	Conteggio del numero di file nei file di origine.
source_file_bytes	bigint	Il numero di byte nei file di origine.
file_count_scanned	integer	Il numero di file sottoposti a scansione da Amazon S3.
file_bytes_scanned	bigint	Il numero di byte sottoposti a scansione da Amazon S3.
error_count	bigint	Conteggio del numero di errori.
copy_job_id	bigint	Identificatore del processo di copia. Il valore `0` indica l'assenza del processo.

Query di esempio

La seguente query mostra le righe caricate, i byte, le tabelle e l'origine dei dati di specifici comandi di copia.

```
SELECT query_id,
       table_name,
       data_source,
       loaded_rows,
       loaded_bytes
FROM sys_load_history
WHERE query_id IN (6389,490791,441663,74374,72297)
ORDER BY query_id,
         data_source DESC;
```

Output di esempio.

```
 query_id |    table_name    |                                data_source
              | loaded_rows | loaded_bytes
----------+------------------
+-----------------------------------------------------------------------+-------------
+---------------
     6389 | store_returns    | s3://load-test/data-sources/tpcds/2.8.0/textfile/1T/
store_returns/    |   287999764 | 1196240296158
    72297 | web_site         | s3://load-test/data-sources/tpcds/2.8.0/textfile/1T/
web_site/         |          54 |         43808
    74374 | ship_mode        | s3://load-test/data-sources/tpcds/2.8.0/textfile/1T/
ship_mode/        |          20 |          1320
   441663 | income_band      | s3://load-test/data-sources/tpcds/2.8.0/textfile/1T/
income_band/      |          20 |          2152
   490791 | customer_address | s3://load-test/data-sources/tpcds/2.8.0/textfile/1T/
customer_address/ |     6000000 |     722924305
```

La seguente query mostra le righe caricate, i byte, le tabelle e l'origine dati dei comandi di copia.

```
SELECT query_id,
       table_name,
       data_source,
       loaded_rows,
       loaded_bytes
FROM sys_load_history
ORDER BY query_id DESC
LIMIT 10;
```

Output di esempio.

```
 query_id |       table_name       |                                    data_source
                          | loaded_rows |  loaded_bytes
----------+------------------------
+-------------------------------------------------------------------------------
+-------------+-----------------
   491058 | web_site               | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/web_site/               |          54 |           43808
   490947 | web_sales              | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/web_sales/              |   720000376 |  22971988122819
   490923 | web_returns            | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/web_returns/            |    71997522 |     96597496325
```

```
   490918 | web_page                | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/web_page/                |        3000 |            1320
   490907 | warehouse               | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/warehouse/               |          20 |            1320
   490902 | time_dim                | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/time_dim/                |       86400 |            1320
   490876 | store_sales             | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/store_sales/             |  2879987999 | 151666241887933
   490870 | store_returns           | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/store_returns/           |   287999764 |   1196405607941
   490865 | store                   | s3://load-test/data-sources/tpcds/2.8.0/
textfile/1T/store/                   |        1002 |          365507
```

La seguente query mostra le righe caricate quotidianamente e i byte del comando di copia.

```
SELECT date_trunc('day',start_time) AS exec_day,
       SUM(loaded_rows) AS loaded_rows,
       SUM(loaded_bytes) AS loaded_bytes
FROM sys_load_history
GROUP BY exec_day
ORDER BY exec_day DESC;
```

Output di esempio.

```
      exec_day       | loaded_rows |   loaded_bytes
---------------------+-------------+------------------
 2022-01-20 00:00:00 |  6347386005 |  258329473070606
 2022-01-19 00:00:00 | 19042158015 |  775198502204572
 2022-01-18 00:00:00 | 38084316030 | 1550294469446883
 2022-01-17 00:00:00 | 25389544020 | 1033271084791724
 2022-01-16 00:00:00 | 19042158015 |  775222736252792
 2022-01-15 00:00:00 | 19834245387 |  798122849155598
 2022-01-14 00:00:00 | 75376544688 | 3077040926571384
```

SYS_MV_REFRESH_HISTORY

I risultati includono informazioni sulla cronologia degli aggiornamenti di tutte le viste materializzate. I risultati includono il tipo di aggiornamento, ad esempio manuale o automatico, e lo stato dell'aggiornamento più recente.

SYS_MV_REFRESH_HISTORY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'identificatore dell'utente che ha inviato l'aggiornamento.
session_id	integer	L'identificatore del processo che esegue l'aggiornamento della vista materializzata.
transaction_id	bigint	L'identificativo della transazione.
database_name	char(128)	Il database che contiene la vista materializzata.
schema_name	char(128)	Lo schema della vista materializzata.
mv_id	bigint	ID oggetto della vista materializzata.
mv_name	char(128)	Il nome della vista materializzata.
refresh_type	char(32)	Il tipo di aggiornamento, ad esempio manuale o automatico.
status	text	Lo stato della richiesta. Per informazioni dettagliate sugli stati, consulta la colonna dello stato per SVL_MV_REFRESH_STATUS.

Nome colonna	Tipo di dati	Descrizione
start_time	timestamp	L'ora di inizio dell'aggi ornamento.
end_time	timestamp	L'ora di fine dell'aggi ornamento.
durata	bigint	La quantità di tempo in microsecondi necessaria per aggiornare la vista materiali zzata.

Query di esempio

La seguente query mostra la cronologia degli aggiornamenti per le viste materializzate.

```
SELECT user_id,
     session_id,
     transaction_id,
     database_name,
     schema_name,
     mv_id,
     mv_name,
     refresh_type,
     status,
     start_time,
     end_time,
     duration
     from sys_mv_refresh_history;
```

Questa query restituisce il seguente output di esempio:

```
 user_id | session_id | transaction_id | database_name | schema_name                |
 mv_id  |  mv_name           |  refresh_type  |  status
         |  start_time                |  end_time                  |  duration
---------+------------+----------------+---------------
+----------------------------+--------+--------------------+----------------
+-----------------------------------------------+---------------------------
+----------------------------+-----------
```

```
      1 | 1073815659 |          15066 | dev           | test_stl_mv_refresh_schema |
 203762 | mv_incremental     | Manual         | MV was already updated
        | 2023-10-26 15:59:20.952179 | 2023-10-26 15:59:20.952866 |      687
      1 | 1073815659 |          15068 | dev           | test_stl_mv_refresh_schema |
 203771 | mv_nonincremental  | Manual         | MV was already updated
        | 2023-10-26 15:59:21.008049 | 2023-10-26 15:59:21.008658 |      609
      1 | 1073815659 |          15070 | dev           | test_stl_mv_refresh_schema |
 203779 | mv_refresh_error   | Manual         | MV was already updated
        | 2023-10-26 15:59:21.064252 | 2023-10-26 15:59:21.064885 |      633
      1 | 1073815659 |          15074 | dev           | test_stl_mv_refresh_schema
 | 203762 | mv_incremental     | Manual         | Refresh successfully updated MV
 incrementally   | 2023-10-26 15:59:29.693329 | 2023-10-26 15:59:43.482842 | 13789513
      1 | 1073815659 |          15076 | dev           | test_stl_mv_refresh_schema |
 203771 | mv_nonincremental  | Manual         | Refresh successfully recomputed MV from
 scratch | 2023-10-26 15:59:43.550184 | 2023-10-26 15:59:47.880833 |  4330649
      1 | 1073815659 |          15078 | dev           | test_stl_mv_refresh_schema |
 203779 | mv_refresh_error   | Manual         | Refresh failed due to an internal error
        | 2023-10-26 15:59:47.949052 | 2023-10-26 15:59:52.494681 |  4545629
(6 rows)
```

SYS_MV_STATE

I risultati includono informazioni sullo stato di tutte le viste materializzate. Include informazioni sulla tabella di base, proprietà dello schema e informazioni sugli eventi recenti, come l'eliminazione di una colonna.

STL_MV_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	bigint	L'ID dell'utente che ha creato l'evento.
transaction_id	bigint	L'ID di transazione dell'evento.
database_name	char(128)	Il database che contiene la vista materializzata.

Nome colonna	Tipo di dati	Descrizione
event_desc	char(500)	L'evento che ha richiesto la modifica dello stato. Esempi di valori possibili sono: • Il tipo di colonna è stato modificato • La colonna è stata eliminata • La colonna è stata rinominata • Il nome dello schema è stato modificato • Conversione piccola tabella • TRUNCATE • Vacuum Nota che ci sono altri valori possibili per questa colonna.
start_time	timestamp	L'orario di inizio dell'evento.
base_table_database_name	char(128)	Il nome del database per la tabella di base.
base_table_schema	char(128)	Lo schema della tabella di base.
base_table_name	char(128)	Il nome della tabella di base.
mv_schema	char(128)	Lo schema della vista materializzata.
mv_name	char(128)	Il nome della vista materializzata.

Nome colonna	Tipo di dati	Descrizione
stato	character(32)	Lo stato modificato della vista materializzata come descritto di seguito: • Ricalcola • Non aggiornabile

Query di esempio

La seguente query mostra lo stato vista materializzata.

```
select * from sys_mv_state;
```

Questa query restituisce il seguente output di esempio:

```
 user_id | transaction_id | database_name | event_desc                  | start_time
              | base_table_database_name | base_table_schema | base_table_name     |
 mv_schema  | mv_name                    | state
---------+----------------+---------------+-----------------------------
+--------------------------+-------------------------+------------------
+---------------------+-------------+---------------------------+--------------
 106     | 12720          | tickit_db     | TRUNCATE                    | 2023-07-26
 14:59:12.788268 | tickit_db                | mv_schema         | test_table_95d6d861 |
 mv_schema   | materialized_view_a1f3f862 | Recompute
 106     | 12724          | tickit_db     | ALTER TABLE ALTER DISTSTYLE | 2023-07-26
 14:59:51.409014 | tickit_db                | mv_schema         | test_table_58102435 |
 mv_schema   | materialized_view_ca746631 | Recompute
 106     | 12720          | tickit_db     | Column was renamed          | 2023-07-26
 14:59:12.822928 | tickit_db                | mv_schema         | test_table_95d6d861 |
 mv_schema   | materialized_view_5750a8d4 | Unrefreshable
 106     | 12727          | tickit_db     | Table was renamed           | 2023-07-26
 15:00:08.051244 | tickit_db                | mv_schema         | test_table_95d6d861 |
 mv_schema   | materialized_view_5750a8d4 | Unrefreshable
 106     | 12720          | tickit_db     | Column was renamed          | 2023-07-26
 14:59:12.857755 | tickit_db                | mv_schema         | test_table_95d6d861 |
 mv_schema   | materialized_view_5750a8d4 | Unrefreshable
 106     | 12727          | tickit_db     | Table was renamed           | 2023-07-26
 15:00:08.051358 | tickit_db                | mv_schema         | test_table_95d6d861 |
 mv_schema   | materialized_view_5ef0d754 | Unrefreshable
```

```
106     | 12720          | tickit_db     | TRUNCATE                    | 2023-07-26
14:59:12.788159 | tickit_db                 | mv_schema         | test_table_95d6d861 |
mv_schema   | materialized_view_5750a8d4 | Recompute
106     | 12720          | tickit_db     | Column was renamed          | 2023-07-26
14:59:12.857799 | tickit_db                 | mv_schema         | test_table_95d6d861 |
mv_schema   | materialized_view_a1f3f862 | Unrefreshable
106     | 12720          | tickit_db     | TRUNCATE                    | 2023-07-26
14:59:12.788327 | tickit_db                 | mv_schema         | test_table_95d6d861 |
mv_schema   | materialized_view_5ef0d754 | Recompute
106     | 12727          | tickit_db     | ALTER TABLE ALTER SORTKEY   | 2023-07-26
15:00:08.006235 | tickit_db                 | mv_schema         | test_table_58102435 |
mv_schema   | materialized_view_ca746631 | Recompute
106     | 12720          | tickit_db     | Column was renamed          | 2023-07-26
14:59:12.82297  | tickit_db                 | mv_schema         | test_table_95d6d861 |
mv_schema   | materialized_view_a1f3f862 | Unrefreshable
106     | 12727          | tickit_db     | Table was renamed           | 2023-07-26
15:00:08.051321 | tickit_db                 | mv_schema         | test_table_95d6d861 |
mv_schema   | materialized_view_a1f3f862 | Unrefreshable
```

SYS_PROCEDURE_CALL

Puoi utilizzare la vista SYS_PROCEDURE_CALL per ottenere informazioni sulle chiamate delle stored procedure, tra cui l'ora di fine, lo stato di una chiamata di una stored procedure e la gerarchia delle chiamate delle stored procedure archiviate. Ogni chiamata di procedura archiviata riceve un'ID query.

SYS_PROCEDURE_CALL è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
session_user_id	integer	L'ID dell'utente che ha creato la sessione ed è l'invoker della chiamata della stored procedure archiviata di primo livello.

Nome colonna	Tipo di dati	Descrizione
security_user_id	integer	L'identificativo dell'utente i cui privilegi erano stati utilizzati per eseguire l'istruzione nella stored procedure. Se la stored procedure è DEFINER, questo sarà lo user_id del proprietario della stored procedure.
query_id	integer	L'identificativo della query della chiamata della stored procedure.
query_text	char(4000)	Il testo della query della chiamata della stored procedure.
start_time	timestamp	L'orario in UTC in cui è iniziata l'esecuzione della query. Il timestamp utilizza sei cifre di precisione per frazioni di secondo, ad esempio. 2009-06-12 11:29:19.131358.
end_time	timestamp	L'orario in UTC in cui è terminata l'esecuzione della query. Il timestamp utilizza sei cifre di precisione per frazioni di secondo, ad esempio: 2009-06-12 11:29:19.131358.

Nome colonna	Tipo di dati	Descrizione
status	char(10)	Lo stato della chiamata alla stored procedure. Quando la stored procedure viene arrestata dal sistema o annullata dall'utente, il valore viene annullato. Se la chiamata alla stored procedure viene eseguita fino al completamento, il valore è operazione riuscita.
caller_procedure_query_id	integer	Se la chiamata alla stored procedure è stata richiamata da un'altra chiamata di stored procedure, questa colonna contiene l'ID query della chiamata esterna. Altrimenti il campo è NULL.

Query di esempio

La seguente query restituisce una gerarchia di chiamata di stored procedure nidificata.

```
select query_id, datediff(seconds, start_time, end_time) as elapsed_time, status,
 trim(query_text) as call, caller_procedure_query_id from sys_procedure_call;
```

Output di esempio.

```
 query_id | elapsed_time | status  |                      call                      |
 caller_procedure_query_id
----------+--------------+---------+------------------------------------------------
+---------------------------
     3087 |           18 | success | CALL proc_bd906c98c45443ffa165e9552056902d(1)    |
          3085
     3085 |           18 | success | CALL proc_bd906c98c45443ffa165e9552056902d_2(1); |
```

```
(2 rows)
```

SYS_PROCEDURE_MESSAGES

SYS_PROCEDURE_MESSAGES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
transaction_id	bigint	L'identificativo della transazione.
query_id	integer	L'identificativo della query della chiamata della stored procedure.
record_time	timestamp	L'ora in UTC in cui il messaggio è stato generato.
log_level	char(10)	Il livello di log del messaggio generato. I valori possibili sono: LOG, INFO, NOTICE, WARNING ed EXCEPTION.
message	char(1024)	Il testo del messaggio generato.
line_number	integer	Il numero di riga del messaggio generato.

Query di esempio

La seguente query mostra un esempio di output di SYS_PROCEDURE_MESSAGES.

```
select transaction_id, query_id, record_time, log_level, trim(message), line_number
 from sys_procedure_messages;
```

```
transaction_id | query_id |        record_time         | log_level |           btrim
        | line_number
---------------+----------+----------------------------+-----------
+--------------------------+-------------
     25267     |   80562  | 2023-07-17 14:38:31.910136 |   NOTICE  |
 test_notice_msg_b9f1e749  |     8
     25267     |   80562  | 2023-07-17 14:38:31.910002 |    LOG    |
 test_log_msg_833c7420     |     6
     25267     |   80562  | 2023-07-17 14:38:31.910111 |    INFO   |
 test_info_msg_651373d9    |     7
     25267     |   80562  | 2023-07-17 14:38:31.910154 |   WARNING |
 test_warning_msg_831c5747 |     9
(4 rows)
```

SYS_QUERY_DETAIL

Utilizzare SYS_QUERY_DETAIL per visualizzare i dettagli delle query a livello di fase. Ogni riga rappresenta un passaggio da una particolare query WLM con dettagli. Questa vista contiene molti tipi di query come DDL, DML e comandi di utilità (ad esempio, copia e scarica). Alcune colonne potrebbero non essere rilevanti a seconda del tipo di query. Ad esempio, external_scanned_bytes non è rilevante per le tabelle interne.

SYS_QUERY_DETAIL è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	Identificatore dell'utente che ha inviato la query.
query_id	bigint	L'identificativo della query.

Nome colonna	Tipo di dati	Descrizione
child_query_sequence	integer	La sequenza della query utente riscritta, che inizia con 1.
stream_id	integer	L'identificatore di flusso del flusso di query.
segment_id	integer	L'identificatore di segmento del segmento di esecuzione della query.
step_id	integer	L'identificatore di passaggio in un segmento.
step_name	text	Il nome della fase in un segmento. I valori possibili sono `aggregate broadcast ,delete,distribut e ,hash,hashjoin, insertlimit,merge,nestloop,pars sortlimit unique, ewindow`.
table_id	integer	L'identificatore della tabella per le scansioni permanenti della tabella.
table_name	character(136)	Nome della tabella del passaggio in corso.
is_rrscan	carattere	Un valore che indica se è una fase di scansione. True (t) indica che è stata utilizzata la scansione a intervallo limitato.

Nome colonna	Tipo di dati	Descrizione
start_time	timestamp	Il momento in cui è iniziata la fase di query.
end_time	timestamp	Il momento in cui la fase di query è stata completata.
durata	bigint	La quantità di tempo (microsec ondi) dedicato alla fase.
avviso	text	La descrizione dell'evento di avviso.
input_bytes	bigint	I byte di input per il passaggio corrente.
input_rows	bigint	Le righe di input per il passaggio corrente.
output_bytes	bigint	I byte di output per il passaggio corrente.
output_rows	bigint	Le righe di output per il passaggio corrente.
blocks_read	bigint	Il numero di blocchi letti dal passaggio.
blocks_write	bigint	Il numero di blocchi scritti dal passaggio.
local_read_IO	bigint	Il numero di blocchi letti dalla cache del disco locale.
remote_read_IO	bigint	Il numero di blocchi letti da remoto.

Nome colonna	Tipo di dati	Descrizione
source	text	Il tipo di oggetto di database scansionato. Questa colonna ha un valore solo quando il valore step_name della riga è `scan`.
data_skewness	integer	L'asimmetria della distribuzione delle righe di output tra tutte le fasi. È un numero compreso tra 0% e 100%. Più alto è il numero, più è sbilanciata la distribuzione.
time_skewness	integer	L'asimmetria del tempo di esecuzione tra tutte le fasi. È un numero compreso tra 0% e 100%. Più alto è il numero, più è sbilanciata la distribuzione.
is_active	carattere	Lo stato della query a livello di fase. I valori possibili sono "t" che indica che la fase è in esecuzione al momento o "f" che indica che la fase è completata.
spilled_block_local_disk	bigint	Il numero di blocchi riversati sul disco locale.
spilled_block_remote_disk	bigint	Il numero di blocchi riversati su Amazon Simple Storage Service.

Nome colonna	Tipo di dati	Descrizione
step_attribute	character(64)	Contiene informazioni sulla fase associata. Valori possibili per le fasi di scansione: `multi-dimensional` .

Query di esempio

L'esempio seguente restituisce l'output di SYS_QUERY_DETAIL.

La seguente query mostra i dettagli dei metadati della query a livello di fase, inclusi nome fase, input_bytes, output_bytes, input_rows, output_rows.

```
SELECT query_id,
       child_query_sequence,
       stream_id,
       segment_id,
       step_id,
       trim(step_name) AS step_name,
       duration,
       input_bytes,
       output_bytes,
       input_rows,
       output_rows
FROM sys_query_detail
WHERE query_id IN (193929)
ORDER BY query_id,
         stream_id,
         segment_id,
         step_id DESC;
```

Output di esempio.

```
 query_id | child_query_sequence | stream_id | segment_id | step_id | step_name  |
 duration     | input_bytes | output_bytes | input_rows | output_rows
----------+----------------------+-----------+------------+---------+------------
+-----------------+-------------+--------------+------------+-------------
   193929 |                    2 |         0 |          0 |       3 | hash       |
      37144 |           0 |      9350272 |          0 |      292196
```

```
193929 |                   5 |         0 |          0 |       3 | hash       |
    9492 |           0 |        23360 |          0 |        1460
193929 |                   1 |         0 |          0 |       3 | hash       |
   46809 |           0 |      9350272 |          0 |      292196
193929 |                   4 |         0 |          0 |       2 | return     |
    7685 |           0 |          896 |          0 |         112
193929 |                   1 |         0 |          0 |       2 | project    |
   46809 |           0 |            0 |          0 |      292196
193929 |                   2 |         0 |          0 |       2 | project    |
   37144 |           0 |            0 |          0 |      292196
193929 |                   5 |         0 |          0 |       2 | project    |
    9492 |           0 |            0 |          0 |        1460
193929 |                   3 |         0 |          0 |       2 | return     |
   11033 |           0 |        14336 |          0 |         112
193929 |                   2 |         0 |          0 |       1 | project    |
   37144 |           0 |            0 |          0 |      292196
193929 |                   1 |         0 |          0 |       1 | project    |
   46809 |           0 |            0 |          0 |      292196
193929 |                   5 |         0 |          0 |       1 | project    |
    9492 |           0 |            0 |          0 |        1460
193929 |                   3 |         0 |          0 |       1 | aggregate  |
   11033 |           0 |       201488 |          0 |          14
193929 |                   4 |         0 |          0 |       1 | aggregate  |
    7685 |           0 |        28784 |          0 |          14
193929 |                   5 |         0 |          0 |       0 | scan       |
    9492 |           0 |        23360 |     292196 |        1460
193929 |                   4 |         0 |          0 |       0 | scan       |
    7685 |           0 |         1344 |        112 |         112
193929 |                   2 |         0 |          0 |       0 | scan       |
   37144 |           0 |      7304900 |     292196 |      292196
193929 |                   3 |         0 |          0 |       0 | scan       |
   11033 |           0 |        13440 |        112 |         112
193929 |                   1 |         0 |          0 |       0 | scan       |
   46809 |           0 |      7304900 |     292196 |      292196
193929 |                   5 |         0 |          0 |      -1 |            |
    9492 |       12288 |            0 |          0 |          0
193929 |                   1 |         0 |          0 |      -1 |            |
   46809 |       16384 |            0 |          0 |          0
193929 |                   2 |         0 |          0 |      -1 |            |
   37144 |       16384 |            0 |          0 |          0
193929 |                   4 |         0 |          0 |      -1 |            |
    7685 |       28672 |            0 |          0 |          0
```

```
   193929 |                   3 |         0 |          0 |      -1 |           |
      11033 |      114688 |            0 |          0 |           0
```

Per visualizzare le tabelle del database nell'ordine dalla più utilizzata a quella meno utilizzata, utilizza l'esempio seguente. Sostituisci *sample_data_dev* con il tuo database. Tieni presente che questa query calcolerà le query a partire dalla creazione del cluster, ma i dati della vista di sistema non vengono salvati quando nel data warehouse manca spazio.

```
SELECT table_name, COUNT (DISTINCT query_id)
FROM SYS_QUERY_DETAIL
WHERE table_name LIKE 'sample_data_dev%'
GROUP BY table_name
ORDER BY COUNT(*) DESC;

+--------------------------------+-------+
|           table_name           | count |
+--------------------------------+-------+
| sample_data_dev.tickit.venue   |     4 |
| sample_data_dev.myunload1.venue |     3 |
| sample_data_dev.tickit.listing  |     1 |
| sample_data_dev.tickit.category |     1 |
| sample_data_dev.tickit.users    |     1 |
| sample_data_dev.tickit.date     |     1 |
| sample_data_dev.tickit.sales    |     1 |
| sample_data_dev.tickit.event    |     1 |
+--------------------------------+-------+
```

SYS_QUERY_HISTORY

Utilizzare SYS_QUERY_HISTORY per visualizzare i dettagli delle query utente. Ogni riga rappresenta una query utente con statistiche accumulate per alcuni campi. Questa visualizzazione contiene molti tipi di query, come DDL (Data Definition Language), DML (Data Manipolation Language), copia, scarico e Amazon Redshift Spectrum. Contiene query sia in esecuzione che finite.

SYS_QUERY_HISTORY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	Identificatore dell'utente che ha inviato la query.
query_id	bigint	L'identificativo della query.
query_label	character(320)	Nome breve per la query.
transaction_id	bigint	L'identificativo della transazione.
session_id	integer	L'identificatore processo del processo che esegue la query.
database_name	character(128)	Il nome del database al quale l'utente era collegato al momento del rilascio della query.
query_type	character(32)	Il tipo di query, ad esempio SELECT, INSERT, UPDATE, UNLOAD, COPY, COMMAND, DDL, UTILITY, CTAS e OTHER.
status	character(10)	Lo stato della query. Valori validi: pianificazione, coda, esecuzione, restituzione, fallito, annullato e riuscito.
result_cache_hit	Booleano	Indica se la query corrisponde alla cache dei risultati.
start_time	timestamp	Il momento in cui è iniziata la query.

Nome colonna	Tipo di dati	Descrizione
end_time	timestamp	L'ora in cui è stata completata la query.
elapsed_time	bigint	La quantità totale di tempo (microsecondi) dedicato alla query.
queue_time	bigint	Il tempo totale (microsecondi) trascorso nella coda di query della classe di servizio.
execution_time	bigint	Tempo totale (microsecondi) in esecuzione nella classe di servizio.
error_message	character(512)	Il motivo per cui una query ha fallito.
returned_rows	bigint	Il numero di righe restituite al client.
returned_bytes	bigint	Il numero di byte restituiti al client.
query_text	character(4000)	Stringa query. Questa stringa potrebbe essere troncata.
redshift_version	character(256)	La versione di Amazon Redshift all'esecuzione della query.
usage_limit	character(150)	Elenco degli ID di limite di utilizzo raggiunti dalla query.

Nome colonna	Tipo di dati	Descrizione
compute_type	varchar(32)	Indica se la query è stata eseguita nel cluster principal e o in un cluster di dimension amento della simultaneità. I valori possibili sono `primary` (la query viene eseguita sul cluster principale), `secondary` (la query viene eseguita sul cluster secondari o) o `primary-scale` (la query viene eseguita sul cluster di simultaneità). Questo è applicabile solo al cluster con provisioning.
compile_time	bigint	Il tempo totale (in microseco ndi) trascorso nella compilazi one della query.
planning_time	bigint	Il tempo totale (in microseco ndi) trascorso nella pianifica zione della query.
lock_wait_time	bigint	Il tempo totale (in microseco ndi) trascorso in attesa del blocco della relazione.

Query di esempio

La seguente query restituisce query in esecuzione e in coda.

```
SELECT user_id,
       query_id,
       transaction_id,
       session_id,
       status,
```

```
       trim(database_name) AS database_name,
       start_time,
       end_time,
       result_cache_hit,
       elapsed_time,
       queue_time,
       execution_time
FROM sys_query_history
WHERE status IN ('running','queued')
ORDER BY start_time;
```

Output di esempio.

```
 user_id | query_id | transaction_id | session_id | status  | database_name |
 start_time         |          end_time          | result_cache_hit | elapsed_time |
 queue_time | execution_time
---------+----------+----------------+------------+---------+---------------
+---------------------------+---------------------------+-----------------
+--------------+------------+----------------
     101 |   760705 |         852337 | 1073832321 | running | tpcds_1t      |
 2022-02-15 19:03:19.67849 | 2022-02-15 19:03:19.739811 | f               |
 61321 |          0 |              0
```

La seguente query restituisce l'ora di inizio, l'ora di fine, il tempo di accodamento, il tempo trascorso, il tempo di pianificazione e altri metadati per una query specifica.

```
SELECT user_id,
       query_id,
       transaction_id,
       session_id,
       status,
       trim(database_name) AS database_name,
       start_time,
       end_time,
       result_cache_hit,
       elapsed_time,
       queue_time,
       execution_time,
       planning_time,
       trim(query_text) as query_text
FROM sys_query_history
WHERE query_id = 3093;
```

Output di esempio.

```
user_id | query_id | transaction_id | session_id |   status   | database_name |
   start_time         |          end_time          | result_cache_hit | elapsed_time |
 queue_time | execution_time | planning_time | query_text
--------+----------+----------------+------------+------------
+---------------+----------------------------+----------------------------
+-----------------+-------------+-----------+---------------+--------------
+-------------------------------------
    106 |     3093 |          11759 | 1073750146 | success    | dev           |
 2023-03-16 16:53:17.840214 | 2023-03-16 16:53:18.106588 | f               |
 266374 |          0 |         105725 |        136589 | select count(*) from item;
```

La seguente query elenca le dieci query SELECT più recenti.

```
SELECT query_id,
       transaction_id,
       session_id,
       start_time,
       elapsed_time,
       queue_time,
       execution_time,
       returned_rows,
       returned_bytes
FROM sys_query_history
WHERE query_type = 'SELECT'
ORDER BY start_time DESC limit 10;
```

Output di esempio.

```
 query_id | transaction_id | session_id |         start_time         | elapsed_time |
 queue_time | execution_time | returned_rows | returned_bytes
----------+----------------+------------+----------------------------+--------------
+------------+----------------+---------------+----------------
   526532 |          61093 | 1073840313 | 2022-02-09 04:43:24.149603 |       520571 |
        0 |         481293 |             1 |           3794
   526520 |          60850 | 1073840313 | 2022-02-09 04:38:27.24875  |       635957 |
        0 |         596601 |             1 |           3679
   526508 |          60803 | 1073840313 | 2022-02-09 04:37:51.118835 |       563882 |
        0 |         503135 |             5 |          17216
   526505 |          60763 | 1073840313 | 2022-02-09 04:36:48.636224 |       649337 |
        0 |         589823 |             1 |            652
```

```
   526478 |          60730 | 1073840313 | 2022-02-09 04:36:11.741471 |     14611321 |
        0 |       14544058 |             0 |              0
   526467 |          60636 | 1073840313 | 2022-02-09 04:34:11.91463  |     16711367 |
        0 |       16633767 |             1 |            575
   511617 |         617946 | 1074009948 | 2022-01-20 06:21:54.44481  |      9937090 |
        0 |        9899271 |           100 |          12500
   511603 |         617941 | 1074259415 | 2022-01-20 06:21:45.71744  |      8065081 |
        0 |        7582500 |           100 |           8889
   511595 |         617935 | 1074128320 | 2022-01-20 06:21:44.030876 |      1051270 |
        0 |        1014879 |             1 |             72
   511584 |         617931 | 1074030019 | 2022-01-20 06:21:42.764088 |       609033 |
        0 |         485887 |           100 |           8438
```

La seguente query mostra il conteggio giornaliero delle query di selezione e il tempo medio trascorso delle query.

```
SELECT date_trunc('day',start_time) AS exec_day,
       status,
       COUNT(*) AS query_cnt,
       AVG(datediff (microsecond,start_time,end_time)) AS elapsed_avg
FROM sys_query_history
WHERE query_type = 'SELECT'
AND start_time >= '2022-01-14'
AND start_time <= '2022-01-18'
GROUP BY exec_day,
         status
ORDER BY exec_day,
         status;
```

Output di esempio.

```
      exec_day       | status  | query_cnt | elapsed_avg
---------------------+---------+-----------+------------
 2022-01-14 00:00:00 | success |      5253 |  56608048
 2022-01-15 00:00:00 | success |      7004 |  56995017
 2022-01-16 00:00:00 | success |      5253 |  57016363
 2022-01-17 00:00:00 | success |      5309 |  55236784
 2022-01-18 00:00:00 | success |      8092 |  54355124
```

La seguente query mostra le prestazioni del tempo trascorso della query giornaliera.

```
SELECT distinct date_trunc('day',start_time) AS exec_day,
```

```
       query_count.cnt AS query_count,
       Percentile_cont(0.5) within group(ORDER BY elapsed_time) OVER (PARTITION BY
 exec_day) AS P50_runtime,
       Percentile_cont(0.8) within group(ORDER BY elapsed_time) OVER (PARTITION BY
 exec_day) AS P80_runtime,
       Percentile_cont(0.9) within group(ORDER BY elapsed_time) OVER (PARTITION BY
 exec_day) AS P90_runtime,
       Percentile_cont(0.99) within group(ORDER BY elapsed_time) OVER (PARTITION BY
 exec_day) AS P99_runtime,
       Percentile_cont(1.0) within group(ORDER BY elapsed_time) OVER (PARTITION BY
 exec_day) AS max_runtime
FROM sys_query_history
LEFT JOIN (SELECT  date_trunc('day',start_time) AS day, count(*) cnt
           FROM sys_query_history
           WHERE query_type = 'SELECT'
           GROUP by 1) query_count
ON date_trunc('day',start_time) = query_count.day
WHERE query_type = 'SELECT'
ORDER BY exec_day;
```

Output di esempio.

```
      exec_day       | query_count | p50_runtime | p80_runtime | p90_runtime |
 p99_runtime  | max_runtime
--------------------+-------------+-------------+-------------+-------------
+-------------+-------------
 2022-01-14 00:00:00 |        5253 |  16816922.0 |  69525096.0 | 158524917.8 |
 486322477.52 | 1582078873.0
 2022-01-15 00:00:00 |        7004 |  15896130.5 |  71058707.0 | 164314568.9 |
 500331542.07 | 1696344792.0
 2022-01-16 00:00:00 |        5253 |  15750451.0 |  72037082.2 | 159513733.4 |
 480372059.24 | 1594793766.0
 2022-01-17 00:00:00 |        5309 |  15394513.0 |  68881393.2 | 160254700.0 |
 493372245.84 | 1521758640.0
 2022-01-18 00:00:00 |        8092 |  15575286.5 |  68485955.4 | 154559572.5 |
 463552685.39 | 1542783444.0
 2022-01-19 00:00:00 |        5860 |  16648747.0 |  72470482.6 | 166485138.2 |
 492038228.67 | 1693483241.0
 2022-01-20 00:00:00 |        1751 |  15422072.0 |  69686381.0 | 162315385.0 |
 497066615.00 | 1439319739.0
 2022-02-09 00:00:00 |          13 |   6382812.0 |  17616161.6 |  21197988.4 |
 23021343.84 |   23168439.0
```

La seguente query mostra la distribuzione del tipo di query.

```
SELECT query_type,
       COUNT(*) AS query_count
FROM sys_query_history
GROUP BY query_type
ORDER BY query_count DESC;
```

Output di esempio.

```
 query_type | query_count
------------+-------------
 UTILITY    |      134486
 SELECT     |       38537
 DDL        |        4832
 OTHER      |         768
 LOAD       |         768
 CTAS       |         748
 COMMAND    |          92
```

SYS_QUERY_TEXT

Utilizza SYS_QUERY_TEXT per visualizzare il testo di tutte le query. Ogni riga rappresenta il testo della query di un massimo di 4000 caratteri che iniziano con il numero di sequenza 0. Quando l'istruzione di query contiene più di 4000 caratteri, vengono registrate righe aggiuntive per l'istruzione incrementando il numero di sequenza per ogni riga. Questa visualizzazione registra tutto il testo delle query degli utenti, ad esempio DDL, utility, query Amazon Redshift e query basate solo sui nodi principali.

SYS_QUERY_TEXT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	Identificatore dell'utente che ha inviato la query.

Nome colonna	Tipo di dati	Descrizione
query_id	bigint	L'identificativo della query.
transaction_id	bigint	L'identificativo della transazione associato all'istruzione.
session_id	integer	L'identificatore del processo della sessione che esegue la query.
start_time	timestamp	L'orario in cui è iniziata la query.
sequenza	integer	Quando una singola istruzione contiene più di 4000 caratteri, per l'istruzione vengono registrate delle righe aggiuntive. La sequenza 0 è la prima riga, 1 la seconda riga e così via.
text	carattere (4000)	Il testo della query SQL con incrementi di 4000 caratteri. Questo campo potrebbe contenere caratteri speciali come la barra rovesciata (\) e la nuova riga (\n).

Query di esempio

La seguente query restituisce query in esecuzione e in coda.

```
SELECT user_id,
 query_id,
 transaction_id,
 session_id, start_time,
 sequence, trim(text) as text from sys_query_text
```

```
 ORDER BY sequence;
```

Output di esempio.

```
 user_id | query_id | transaction_id | session_id |        start_time          |
 sequence |                                                        text
--------+----------+----------------+------------+----------------------------
+----------
+-----------------------------------------------------------------------------------------------------------------
   100  |     4    |       1396     | 1073750220 | 2023-04-28 16:44:55.887184 |
   0    | SELECT trim(text) as text, sequence FROM sys_query_text WHERE query_id =
 pg_last_query_id() AND user_id > 1 AND start
_time > '2023-04-28 16:44:55.922705+00:00'::timestamp order by sequence;
```

La seguente query restituisce le autorizzazioni concesse o revocate ai gruppi del database.

```
SELECT
    SPLIT_PART(text, ' ', 1) as grantrevoke,
    SPLIT_PART((SUBSTRING(text, STRPOS(UPPER(text), 'GROUP'))), ' ', 2) as group,
    SPLIT_PART((SUBSTRING(text, STRPOS(UPPER(text), ' '))), 'ON', 1) as type,
    SPLIT_PART((SUBSTRING(text, STRPOS(UPPER(text), 'ON'))), ' ', 2) || ' ' ||
 SPLIT_PART((SUBSTRING(text, STRPOS(UPPER(text), 'ON'))), ' ', 3) as entity
FROM SYS_QUERY_TEXT
WHERE (text LIKE 'GRANT%' OR text LIKE 'REVOKE%') AND text LIKE '%GROUP%';

+-------------+----------+--------+----------+
| grantrevoke |  group   |  type  |  entity  |
+-------------+----------+--------+----------+
| GRANT       | bi_group | SELECT | TABLE t1 |
| GRANT       | bi_group | SELECT | TABLE t1 |
| GRANT       | bi_group | SELECT | TABLE t1 |
| GRANT       | bi_group | USAGE  | TABLE t1 |
| GRANT       | bi_group | SELECT | TABLE t1 |
| GRANT       | bi_group | SELECT | TABLE t1 |
+-------------+----------+--------+----------+
```

SYS_RESTORE_LOG

Usa SYS_RESTORE_LOG per monitorare l'avanzamento di ogni tabella nel cluster durante il ridimensionamento classico nei nodi RA3. Acquisisce la velocità di trasmissione effettiva storica della migrazione dei dati durante l'operazione di ridimensionamento. Per ulteriori informazioni sul ridimensionamento classico ai nodi RA3, consulta Classic resize (Ridimensionamento classico).

SYS_RESTORE_LOG è visibile solo agli utenti con privilegi avanzati.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
event_time	timestamp	Timestamp che indica quando viene registrata la voce di log.
database_name	char(128)	Nome del database.
schema_name	char(128)	Il nome dello schema.
table_name	char(128)	Nome della tabella.
table_id	integer	L'ID della tabella.
action	char(128)	L'azione intrapresa al momento dell'inserimento. I valori possono essere: migrazione iniziata, checkpoin t, ripresa, completata, annullata o ripristinata.
table_size	Long	La dimensione della tabella
total_data_processed	Long	La dimensione dei dati in MB elaborati fino a questo punto della tabella.
delta_data_processed	Long	La dimensione dei dati elaborati dall'ultimo aggiornam ento di event_time, in MB. Consente di determinare la quantità di dati elaborata rispetto al precedente intervall o di tempo registrato. Puoi confrontare questo valore con table_size per avere un'idea

Nome colonna	Tipo di dati	Descrizione
		della velocità con cui procede l'elaborazione dei dati.
message	har(512)	Spiegazione dettagliata del valore nella colonna delle azioni.
redistribution_type	char(32)	Il tipo di ridistribuzione della tabella. Conversione KEY o operazione di ribilanciamento EVEN. Per ulteriori informazioni sugli stili di distribuzione, consulta Stili di distribuzione.

Query di esempio

La seguente query calcola la velocità effettiva di elaborazione dei dati utilizzando SYS_RESTORE_LOG.

```
SELECT
    ROUND(sum(delta_data_processed) / 1024.0, 2) as data_processed_gb,
    ROUND(datediff(sec, min(event_time), max(event_time)) / 3600.0, 2) as duration_hr,
    ROUND(data_processed_gb/duration_hr, 2) as throughput_gb_per_hr
from sys_restore_log;
```

Output di esempio.

```
 data_processed_gb | duration_hr | throughput_gb_per_hr
-------------------+-------------+----------------------
              0.91 |        8.37 |                 0.11
(1 row)
```

La seguente query che mostra tutti i tipi di ridistribuzione.

```
SELECT * from sys_restore_log ORDER BY event_time;
```

```
 database_name |     schema_name      |      table_name      | table_id |
 action             | total_data_processed | delta_data_processed |         event_time
        | table_size | message |   redistribution_type
---------------+----------------------+----------------------+----------
+----------------------------+----------------------+----------------------
+----------------------------+------------+---------+--------------------------
 dev           | schemaaaa877096d844d | customer_key         |   106424 |
 Redistribution started      |                    0 |                       | 2024-01-05
 02:18:00.744977 |        325 |         | Restore Distkey Table
 dev           | schemaaaa877096d844d | dp30907_t2_autokey   |   106430 |
 Redistribution started      |                    0 |                       | 2024-01-05
 02:18:02.756675 |         90 |         | Restore Distkey Table
 dev           | schemaaaa877096d844d | dp30907_t2_autokey   |   106430 |
 Redistribution completed    |                   90 |                    90 | 2024-01-05
 02:23:30.643718 |         90 |         | Restore Distkey Table
 dev           | schemaaaa877096d844d | customer_key         |   106424 |
 Redistribution completed    |                  325 |                   325 | 2024-01-05
 02:23:45.998249 |        325 |         | Restore Distkey Table
 dev           | schemaaaa877096d844d | dp30907_t1_even      |   106428 |
 Redistribution started      |                    0 |                       | 2024-01-05
 02:23:46.083849 |         30 |         | Rebalance Disteven Table
 dev           | schemaaaa877096d844d | dp30907_t5_auto_even |   106436 |
 Redistribution started      |                    0 |                       | 2024-01-05
 02:23:46.855728 |         45 |         | Rebalance Disteven Table
 dev           | schemaaaa877096d844d | dp30907_t5_auto_even |   106436 |
 Redistribution completed    |                   45 |                    45 | 2024-01-05
 02:24:16.343029 |         45 |         | Rebalance Disteven Table
 dev           | schemaaaa877096d844d | dp30907_t1_even      |   106428 |
 Redistribution completed    |                   30 |                    30 | 2024-01-05
 02:24:20.584703 |         30 |         | Rebalance Disteven Table
 dev           | schemaefd028a2a48a4c | customer_even        |   130512 |
 Redistribution started      |                    0 |                       | 2024-01-05
 04:54:55.641741 |        190 |         | Restore Disteven Table
 dev           | schemaefd028a2a48a4c | customer_even        |   130512 |
 Redistribution checkpointed |     29.4342113157737 |     29.4342113157737 | 2024-01-05
 04:55:04.770696 |        190 |         | Restore Disteven Table
(8 rows)
```

SYS_RESTORE_STATE

Usa SYS_RESTORE_STATE per monitorare l'avanzamento della migrazione di ogni tabella durante un ridimensionamento classico. È applicabile in particolare quando il tipo di nodo di destinazione è

RA3. Per ulteriori informazioni sul ridimensionamento classico ai nodi RA3, consulta Classic resize (Ridimensionamento classico).

SYS_RESTORE_STATE è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	Identificatore dell'utente che ha inviato la query.
database_name	char(64)	Il nome del database della tabella.
schema_id	integer	L'ID dello schema della tabella.
table_id	integer	L'ID della tabella.
table_name	char(128)	Nome della tabella.
redistribution_status	char(128)	Lo stato dell'avanzamento della ridistribuzione della tabella. I valori possibili sono `Completed` , `In progress` e `Pending`.
percentage_redistributed	float	Lo stato dell'avanzamento della ridistribuzione della tabella. I valori possibili vanno da 0 fino a 100%. Ad esempio, il valore 25 indica che il 25% dei dati viene ridistribuito.
redistribution_type	char(32)	Il tipo di ridistribuzione della tabella. Conversione KEY o operazione di ribilanciamento EVEN. Per ulteriori informazi

Nome colonna	Tipo di dati	Descrizione
		oni sugli stili di distribuzione, consulta Stili di distribuzione.

Query di esempio

La seguente query restituisce i record per le interrogazioni in esecuzione e in coda.

```
SELECT * FROM sys_restore_state;
```

Output di esempio.

```
 userid | database_name | schema_id | table_id |   table_name   | redistribution_status
 | precentage_redistributed |   redistribution_type
--------+---------------+-----------+----------+----------------
+-----------------------+--------------------------+-------------------------
    1   |     test1     |   124865  |  124878  | customer_key_4 |         Pending
 |      0                 |  Rebalance Disteven Table
    1   |      dev      |   124865  |  124874  | customer_key_3 |         Pending
 |      0                 |  Rebalance Disteven Table
    1   |      dev      |   124865  |  124870  | customer_key_2 |        Completed
 |     100                |  Rebalance Disteven Table
    1   |      dev      |   124865  |  124866  | customer_key_1 |       In progress
 |     13.52              |  Restore Distkey Table
```

Di seguito viene fornito lo stato dell'elaborazione dei dati.

```
SELECT
    redistribution_status, ROUND(SUM(block_count) / 1024.0, 2) AS total_size_gb
FROM sys_restore_state sys inner join stv_tbl_perm stv
    on sys.table_id = stv.id
GROUP BY sys.redistribution_status;
```

Output di esempio.

```
 redistribution_status | total_size_gb
-----------------------+---------------
 Completed             |          0.07
```

```
 Pending               |          0.71
 In progress           |          0.20
(3 rows)
```

SYS_SCHEMA_QUOTA_VIOLATIONS

Registra occorrenza, ID transazione e altre informazioni utili quando viene superata una quota dello schema. Questa tabella di sistema è una traduzione di STL_SCHEMA_QUOTA_VIOLATIONS.

r_SYS_SCHEMA_QUOTA_VIOLATIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
owner_id	integer	ID del proprietario dello schema.
user_id	integer	L'ID dell'utente che ha generato la voce.
transaction_id	bigint	L'ID di transazione associato all'istruzione.
session_id	integer	L'ID di processo associato all'istruzione.
schema_id	integer	Lo spazio dei nomi o l'ID dello schema.
schema_name	carattere (128)	Lo spazio dei nomi o il nome dello schema.
quota	integer	La quantità di spazio su disco (in MB) che lo schema può utilizzare.
disk_usage	integer	Lo spazio su disco (in MB) attualmente utilizzato dallo schema.
record_time	timestamp without time zone	L'ora in cui si è verificata la violazione.

Query di esempio

La query seguente mostra il risultato della violazione della quota:

```
SELECT user_id, TRIM(schema_name) "schema_name", quota, disk_usage, record_time FROM
sys_schema_quota_violations WHERE SCHEMA_NAME = 'sales_schema' ORDER BY timestamp DESC;
```

Questa query restituisce il seguente output di esempio per lo schema specificato:

```
user_id| schema_name  | quota | disk_usage | record_time
-------+--------------+-------+------------+----------------------------
104    | sales_schema | 2048  | 2798       | 2020-04-20 20:09:25.494723
(1 row)
```

SYS_SERVERLESS_USAGE

Utilizzare SYS_SERVERLESS_USAGE per visualizzare i dettagli dell'utilizzo delle risorse da parte di Amazon Redshift Serverless. Questa vista di sistema non si applica ai cluster Amazon Redshift di cui è stato eseguito il provisioning.

Questa vista contiene il riepilogo dell'utilizzo serverless, inclusa la quantità di capacità di calcolo utilizzata per elaborare le query e la quantità di archiviazione gestita di Amazon Redshift utilizzata con una granularità di 1 minuto. La capacità di calcolo viene misurata in unità di elaborazione Redshift (RPU) per i carichi di lavoro eseguiti in secondi RPU al secondo. Le RPU vengono utilizzate per elaborare query sui dati caricati nel data warehouse, richieste da un data lake Amazon S3 o accessibili da database operativi utilizzando una query federata. Amazon Redshift serverless mantiene le informazioni in SYS_SERVERLESS_USAGE per 7 giorni.

Per esempi sulla fatturazione dei costi di calcolo, consulta Fatturazione per Amazon Redshift serverless.

SYS_SERVERLESS_USAGE è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
start_time	timestamp	Il momento in cui è iniziato l'intervallo.
end_time	timestamp	Il tempo in cui l'intervallo è stato completato.
compute_seconds	double precision	L'unità di calcolo accumulata (RPU) secondi consumati durante questo intervallo di tempo. Questo valore rappresenta la capacità RPU di base allocata per l'account.
compute_capacity	double precision	Il numero medio di unità di calcolo (unità di elaborazione Redshift o RPU) allocate durante questo intervallo di tempo. Il valore compute_capacity può essere modificato dinamicamente.
data_storage	integer	Lo spazio medio di archiviazione di dati (in MB) utilizzato durante questo intervallo di tempo. L'archiviazione dati utilizzata può cambiare dinamicamente man mano che i dati vengono caricati o eliminati dal database.

Nome colonna	Tipo di dati	Descrizione
cross_region_transferred_data	integer	I dati accumulati trasferiti per la condivisione dei dati tra regioni in byte durante questo intervallo di tempo.
charged_seconds	integer	Secondi RPU accumulati e addebitati durante questo intervallo di tempo. Il calcolo viene effettuato una volta terminate le transazioni, per cui può essere 0 durante l'esecuzione di una transazio ne. Utilizza charged_seconds per calcolare il costo per un gruppo di lavoro Amazon Redshift serverless. Questo valore rappresenta la capacità RPU allocata per il gruppo di lavoro Amazon Redshift serverless.

Note per l'utilizzo

- Esistono situazioni in cui compute_seconds è 0 ma charged_seconds è maggiore di 0 o viceversa. Si tratta di un comportamento normale dovuto al modo in cui i dati vengono registrati nella vista di sistema. Per una rappresentazione più accurata dei dettagli sull'utilizzo serverless, consigliamo di aggregare i dati.

Esempio

Per ottenere i costi totali per le ore RPU utilizzate per un intervallo di tempo usando charged_seconds, esegui la query riportata sotto:

```
select trunc(start_time) "Day",
(sum(charged_seconds)/3600::double precision) * <Price for 1 RPU> as cost_incurred
```

```
from sys_serverless_usage
group by 1
order by 1
```

Nota: può esserci un tempo di inattività durante l'intervallo. Il tempo di inattività non viene aggiunto alle RPU consumate.

SYS_SESSION_HISTORY

Utilizza SYS_SESSION_HISTORY per visualizzare le informazioni sulle sessioni attive correnti e sulla cronologia delle sessioni.

SYS_SESSION_HISTORY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	char(50)	L'identificativo dell'utente che ha generato la voce.
session_id	integer	L'ID della sessione associata all'istruzione.
database_name	char(50)	Nome del database.
status	char	Lo stato della sessione. I valori possibili sono `active`, `timed out` e `closed`.
session_timeout	integer	Il tempo massimo in secondi in cui una sessione rimane inattiva o inattiva prima del timeout. 0 indica che non è impostato alcun timeout.
start_time	timestamp	Il timestamp in cui è stata stabilita la connessione.
end_time	timestamp	Il timestamp in cui è stata interrotta la connessione.

Esempio

Di seguito viene riportato un esempio di output di SYS_SESSION_HISTORY.

```
select * from sys_session_history;
 user_id | session_id |   database_name  | status | session_timeout |
 start_time         |          end_time
---------+------------+------------------+--------+-----------------
+----------------------------+----------------------------
       1 | 1073971370 | dev              | closed |        0        | 2023-07-17
 15:50:12.030104 | 2023-07-17 15:50:12.123218
       1 | 1073979694 | dev              | closed |        0        | 2023-07-17
 15:50:24.117947 | 2023-07-17 15:50:24.131859
       1 | 1073873049 | dev              | closed |        0        | 2023-07-17
 15:49:29.067398 | 2023-07-17 15:49:29.070294
       1 | 1073873086 | database18127a4a | closed |        0        | 2023-07-17
 15:49:29.119018 | 2023-07-17 15:49:29.125925
       1 | 1073832112 | dev              | closed |        0        | 2023-07-17
 15:49:29.164688 | 2023-07-17 15:49:29.179631
       1 | 1073987697 | dev              | closed |        0        | 2023-07-17
 15:49:29.26749  | 2023-07-17 15:49:29.273034
       1 | 1073922429 | dev              | closed |        0        | 2023-07-17
 15:49:33.35315  | 2023-07-17 15:49:33.367499
       1 | 1073766783 | dev              | closed |        0        | 2023-07-17
 15:49:45.38237  | 2023-07-17 15:49:45.396902
       1 | 1073807506 | dev              | active |        0        | 2023-07-17
 15:51:48        |
```

SYS_SPATIAL_SIMPLIFY

Puoi eseguire una query nella vista di sistema SYS_SPATIAL_SIMPLIFY per ottenere informazioni sugli oggetti della geometria spaziale semplificata utilizzando il comando COPY. Quando utilizzi COPY in uno shapefile, puoi specificare le opzioni di importazione `tolerance`, SIMPLIFY AUTO, and SIMPLIFY AUTO `max_tolerance`. Il risultato della semplificazione è riepilogato nella vista di sistema SVL_SPATIAL_SIMPLIFY.

Quando si imposta SIMPLIFY AUTO `max_tolerance`, questa vista contiene una riga per ogni geometria che ha superato la dimensione massima. Quando è impostato SIMPLIFY `tolerance`, viene memorizzata una riga per l'intera operazione COPY. Questa riga fa riferimento all'ID query COPY e alla tolleranza di semplificazione specificata.

Per ulteriori informazioni sul caricamento di shapefile, consultare Caricamento di uno shapefile in Amazon Redshift.

SVL_SPATIAL_SIMPLIFY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query_id	bigint	ID della query (comando COPY) che ha generato questa riga.
line_number	bigint	Quando viene specificata l'opzione COPY `SIMPLIFY AUTO`, questo valore è il numero di record del record semplificato nello shapefile.
maximum_tolerance	double precision	Il valore di tolleranza della distanza specificato nel comando COPY. Questo è il valore di tolleranza massima quando si utilizza l'opzione `SIMPLIFY AUTO` oppure il valore di tolleranza fissa quando si utilizza l'opzione `SIMPLIFY`.
initial_size	bigint	La dimensione in byte del valore dei dati `GEOMETRY` prima della semplificazione.
simplified	char(1)	Quando viene specificata l'opzione COPY `SIMPLIFY AUTO`, `t` se la geometria è stata semplificata con successo, oppure `f` in caso contrario. La geometria potrebbe non essere semplificata correttamente se dopo la semplific azione con la tolleranza massima specificata, la sua dimensione è ancora maggiore della dimensione massima della geometria.

Nome colonna	Tipo di dati	Descrizione
final_size	bigint	Quando viene specificata l'opzione COPY SIMPLIFY AUTO, questa è la dimensione in byte della geometria dopo la semplificazione.
final_tolerance	double precision	Tolleranza finale scelta per la semplificazione.

Query di esempio

La seguente query restituisce l'elenco dei record semplificati da COPY.

```
SELECT * FROM sys_spatial_simplify;


 query_id | line_number | maximum_tolerance | initial_size | simplified | final_size |
  final_tolerance
----------+-------------+-------------------+--------------+------------+------------
+----------------------
    20    |     1184704 |                -1 |      1513736 | t          |    1008808 |
  1.276386653895e-05
    20    |     1664115 |                -1 |      1233456 | t          |    1023584 |
 6.11707814796635e-06
```

SYS_STREAM_SCAN_ERRORS

Registra gli errori per i record caricati tramite l'importazione dati in streaming.

SYS_STREAM_SCAN_ERRORS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
external_schema_name	character (128)	Nome del flusso Kinesis o dello schema di un argomento Amazon MSK. Applica la distinzione tra lettere maiuscole e minuscole.
stream_name	character (255)	Nome del flusso o dell'argomento. Applica la distinzione tra lettere maiuscole e minuscole.
mv_name	character (128)	Nome della vista materializzata associata. È vuoto se non è presente un nome. Applica la distinzione tra lettere maiuscole e minuscole.
transaction_id	bigint	L'ID transazione.
query_id	bigint	L'ID di query.
stream_timestamp_type	character (1)	Il tipo di timestamp del flusso. Applica la distinzione tra lettere maiuscole e minuscole.
stream_timestamp	timestamp without time zone	Ora di arrivo del record.
record_time	timestamp without time zone	Ora in cui è stato registrato il messaggio di errore.
partition_id	character (128)	ID della partizione/condivisione. Applica la distinzione tra lettere maiuscole e minuscole.

Nome colonna	Tipo di dati	Descrizione
position	character (128)	Posizione del record. Corrisponde al numero di sequenza in Kinesis o all'offset in Amazon MSK. Applica la distinzione tra lettere maiuscole e minuscole.
error_code	integer	Il codice di errore.
error_reason	character (128)	Motivo dell'errore. Applica la distinzione tra lettere maiuscole e minuscole.

SYS_STREAM_SCAN_STATES

Registra gli stati di scansione dei record caricati tramite l'importazione dati in streaming.

SYS_STREAM_SCAN_STATES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
external_schema_name	character (128)	Nome dello schema esterno. Applica la distinzione tra lettere maiuscole e minuscole.
stream_name	character (255)	Il nome del flusso. Applica la distinzione tra lettere maiuscole e minuscole.
mv_name	character (128)	Nome della vista materializzata associata. È vuoto se non è presente un nome. Applica la distinzione tra lettere maiuscole e minuscole.
transaction_id	bigint	L'ID transazione.

Nome colonna	Tipo di dati	Descrizione
query_id	bigint	L'ID di query.
record_time	timestamp without time zone	Ora della registrazione dei dati.
partition_id	character (128)	ID della partizione. Applica la distinzione tra lettere maiuscole e minuscole.
latest_position	character (128)	Posizione dell'ultimo record letto nel batch. Corrisponde al numero di sequenza in Kinesis o all'offset in Amazon MSK. Applica la distinzione tra lettere maiuscole e minuscole.
scanned_rows	bigint	Numero di record analizzati nel batch.
skipped_rows	bigint	Numero di record che sono stati ignorati nel batch.
scanned_bytes	bigint	Numero di byte analizzati nel batch.
stream_record_time_min	timestamp without time zone	Ora minima di arrivo del primo record del batch in Kinesis o Amazon MSK.
stream_record_time_max	timestamp without time zone	Ora massima di arrivo dell'ultimo record del batch in Kinesis o Amazon MSK.

La seguente query mostra i dati relativi ai flussi e agli argomenti per query specifiche.

```
select
 query_id,mv_name::varchar,external_schema_name::varchar,stream_name::varchar,sum(scanned_rows)
 total_records,
sum(scanned_bytes) total_bytes from sys_stream_scan_states where query in
 (5401180,8601939) group by 1,2,3,4;

  query_id  |    mv_name     | external_schema_name |   stream_name   | total_records |
  total_bytes
------------+----------------+----------------------+-----------------+---------------
+----------------
 5401180    | kinesistest    | kinesis              | kinesisstream   |    1493255696 |
 3209006490704
 8601939    | msktest        | msk                  | mskstream       |      14677023 |
   31056580668
(2 rows)
```

SYS_TRANSACTION_HISTORY

Utilizza SYS_TRANSACTION_HISTORY per visualizzare i dettagli di una transazione durante il monitoraggio di una query.

SYS_TRANSACTION_HISTORY è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	ID dell'utente che ha generato la voce.
transaction_id	bigint	L'ID della transazione.
isolation_level	text	Il livello di isolamento della transazione. I valori possibili sono `Serializable` e `Snapshot Isolation` .

Nome colonna	Tipo di dati	Descrizione
status	text	L'ID della transazione. Gli stati possibili sono `committed` e `rolledback` .
transaction_start_time	timestamp	L'ora di inizio della transazione.
commit_start_time	timestamp	L'orario di inizio del commit.
commit_end_time	timestamp	L'orario di file del commit.
blocks_committed	bigint	Il numero di blocchi scritti come parte di questo commit.
undo_transaction_id	bigint	L'ID della transazione di annullamento se alcune transazioni sono state annullate. In caso contrario, questo valore è -1.

Query di esempio

```
select * from sys_transaction_history order by transaction_start_time desc;

 user_id | transaction_id | isolation_level |   status   |   transaction_start_time
   |     commit_start_time      |      commit_end_time       | blocks_committed |
 undo_transaction_id
---------+----------------+-----------------+------------+--------------------------
+---------------------------+---------------------------+------------------
+---------------------
     100 |           1310 | Serializable    | committed  | 2023-08-27 21:03:11.822205 |
 2023-08-28 21:03:11.825287 | 2023-08-28 21:03:11.854883 |               17 |
        -1
     101 |           1345 | Serializable    | committed  | 2023-08-27 21:03:12.000278 |
 2023-08-28 21:03:12.003736 | 2023-08-28 21:03:12.030061 |               17 |
        -1
```

```
    102 |           1367 | Serializable    | committed  | 2023-08-27 21:03:12.1532   |
2023-08-28 21:03:12.156124 | 2023-08-28 21:03:12.186468 |               17 |
       -1
    100 |           1370 | Serializable    | committed  | 2023-08-27 21:03:12.199316 |
2023-08-28 21:03:12.204854 | 2023-08-28 21:03:12.238186 |               24 |
       -1
    100 |           1408 | Serializable    | committed  | 2023-08-27 21:03:53.891107 |
2023-08-28 21:03:53.894825 | 2023-08-28 21:03:53.927465 |               17 |
       -1
    100 |           1409 | Serializable    | rolledback | 2023-08-27 21:03:53.936431 |
2000-01-01 00:00:00        | 2023-08-28 21:04:08.712532 |                0 |
     1409
    101 |           1415 | Serializable    | committed  | 2023-08-27 21:04:24.283188 |
2023-08-28 21:04:24.289196 | 2023-08-28 21:04:24.374318 |               25 |
       -1
    101 |           1416 | Serializable    | committed  | 2023-08-27 21:04:24.38818  |
2023-08-28 21:04:24.391688 | 2023-08-28 21:04:24.415135 |               17 |
       -1
    100 |           1417 | Serializable    | rolledback | 2023-08-27 21:04:24.424252 |
2000-01-01 00:00:00        | 2023-08-28 21:04:28.354826 |                0 |
     1417
    101 |           1418 | Serializable    | rolledback | 2023-08-27 21:04:24.425195 |
2000-01-01 00:00:00        | 2023-08-28 21:04:28.680355 |                0 |
     1418
    100 |           1420 | Serializable    | committed  | 2023-08-27 21:04:28.697607 |
2023-08-28 21:04:28.702374 | 2023-08-28 21:04:28.735541 |               23 |
       -1
    101 |           1421 | Serializable    | committed  | 2023-08-27 21:04:28.744854 |
2023-08-28 21:04:28.749344 | 2023-08-28 21:04:28.779029 |               23 |
       -1
    101 |           1423 | Serializable    | committed  | 2023-08-27 21:04:28.78942  |
2023-08-28 21:04:28.791556 | 2023-08-28 21:04:28.817485 |               16 |
       -1
    100 |           1430 | Serializable    | committed  | 2023-08-27 21:04:28.917788 |
2023-08-28 21:04:28.919993 | 2023-08-28 21:04:28.944812 |               16 |
       -1
    102 |           1494 | Serializable    | committed  | 2023-08-27 21:04:37.029058 |
2023-08-28 21:04:37.033137 | 2023-08-28 21:04:37.062001 |               16 |
       -1
```

SYS_UDF_LOG

Registra la generazione di messaggi di errore e di avviso definita dal sistema durante l'esecuzione della funzione definita dall'utente (UDF).

SYS_UDF_LOG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query_id	bigint	L'identificativo della query.
function_name	text	Il nome della funzione definita dall'utente.
record_time	timestamp	L'orario in cui è stato creato il record.
sequenza	integer	La sequenza di un singolo messaggio di log.
message	text	Il testo del messaggio di log.

Query di esempio

L'esempio seguente mostra come le UDF gestiscono gli errori definiti dal sistema. Il primo blocco mostra la definizione di una funzione UDF che restituisce l'inverso di un argomento. Quando esegui la funzione e fornisci un argomento 0, la funzione restituisce un errore. L'ultima istruzione restituisce il messaggio di errore registrato in SYS_UDF_LOG.

```
-- Create a function to find the inverse of a number.
CREATE OR REPLACE FUNCTION f_udf_inv(a int)

RETURNS float

IMMUTABLE AS $$return 1/a

$$ LANGUAGE plpythonu;
```

```
-- Run the function with 0 to create an error.
Select f_udf_inv(0);

-- Query SYS_UDF_LOG to view the message.
Select query_id, record_time, message::varchar from sys_udf_log;


query_id    |    record_time              |                 message
----------+----------------------------
+-----------------------------------------------------
2211        | 2023-08-23 15:53:11.360538 |  ZeroDivisionError: integer division or
 modulo by zero line 2, in f_udf_inv\n return 1/a\n
```

L'esempio seguente aggiunge la registrazione e un messaggio di avviso all'UDF in modo che un'operazione di divisione per zero risulti in un messaggio di avviso anziché in un arresto con un messaggio di errore.

```
-- Create a function to find the inverse of a number and log a warning if you input 0.
CREATE OR REPLACE FUNCTION f_udf_inv_log(a int)
  RETURNS float IMMUTABLE
 AS $$
  import logging
  logger = logging.getLogger() #get root logger
  if a==0:
    logger.warning('You attempted to divide by zero.\nReturning zero instead of error.
\n')
    return 0
  else:
     return 1/a
$$ LANGUAGE plpythonu;

-- Run the function with 0 to trigger the warning.
Select f_udf_inv_log(0);

-- Query SYS_UDF_LOG to view the message.
Select query_id, record_time, message::varchar from sys_udf_log;

 query_id |        record_time         |                                  message
----------+----------------------------
+-------------------------------------------------------------------------------
     0   | 2023-08-23 16:10:48.833503 | WARNING: You attempted to divide by zero.
\nReturning zero instead of error.\n
```

SYS_UNLOAD_DETAIL

Utilizzare SYS_UNLOAD_DETAIL per visualizzare i dettagli di un'operazione UNLOAD. Registra una riga per ogni file creato da un'istruzione UNLOAD. Ad esempio, se un'istruzione UNLOAD crea 12 file, SYS_UNLOAD_DETAIL conterrà 12 righe corrispondenti.

SYS_UNLOAD_DETAIL è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'identificativo dell'utente che ha generato la voce.
query_id	integer	L'identificatore della query del comando UNLOAD.
session_id	integer	L'ID del processo associato all'istruzione di query.
transaction_id	bigint	L'ID della transazione associato all'istruzione della query.
file_name	character (1280)	Il percorso completo di un oggetto Amazon S3 per il file.
start_time	timestamp	Il momento in cui è iniziata la transazione.
end_time	timestamp	Il momento in cui è stata completata la transazione.
line_count	bigint	Il numero di righe scaricate nel file.
transfer_size	bigint	Il numero di byte trasferiti.

Nome colonna	Tipo di dati	Descrizione
file_format	character (10)	Il formato dei file non caricati.

Query di esempio

La seguente query mostra i dettagli della query scaricata, inclusi formato, righe e numero di file del comando unload.

```
select query_id, substring(file_name, 0, 50), transfer_size, file_format from
 sys_unload_detail;
```

Output di esempio.

```
 query_id |                    substring                     | transfer_size |
 file_format
----------+--------------------------------------------------+---------------
+-------------
     9272 | s3://my-bucket/my_unload_doc_venue0000_part_00.gz |        395886 | Text

     9272 | s3://my-bucket/my_unload_doc_venue0001_part_00.gz |        406444 | Text

     9272 | s3://my-bucket/my_unload_doc_venue0002_part_00.gz |        409431 | Text

     9272 | s3://my-bucket/my_unload_doc_venue0003_part_00.gz |        403051 | Text

     9272 | s3://my-bucket/my_unload_doc_venue0004_part_00.gz |        413592 | Text

     9272 | s3://my-bucket/my_unload_doc_venue0005_part_00.gz |        395689 | Text

(6 rows)
```

SYS_UNLOAD_HISTORY

Utilizzare SYS_UNLOAD_HISTORY per visualizzare i dettagli dei comandi UNLOAD. Ogni riga rappresenta un comando UNLOAD con statistiche accumulate per alcuni campi. Contiene comandi UNLOAD in esecuzione e finiti.

SYS_UNLOAD_HISTORY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'identificatore dell'utente che ha inviato lo scaricamento.
query_id	bigint	L'identificatore della query del comando UNLOAD.
transaction_id	bigint	L'identificativo della transazione.
session_id	integer	L'identificatore processo del processo che esegue lo scaricamento.
database_name	text	Il nome del database al quale l'utente era collegato al momento del rilascio dell'operazione.
status	text	Stato del comando UNLOAD. I valori validi sono: `running`, `completed` , `aborted` e `unknown`.
start_time	timestamp	Il momento in cui è iniziato lo scaricamento.
end_time	timestamp	Il momento in cui lo scaricamento è stato completato.
durata	bigint	Il tempo trascorso (microsecondi) nel comando UNLOAD.

Nome colonna	Tipo di dati	Descrizione
file_format	text	Il formato di file dei file di output.
compression_type	text	Il tipo di compressione.
unloaded_location	text	La posizione di Amazon S3 dei file scaricati.
unloaded_rows	bigint	Il numero di righe.
unloaded_files_count	bigint	Il numero di file del file di output.
unloaded_files_size	bigint	La dimensione del file di output.
error_message	text	Il messaggio di errore del comando UNLOAD.

Query di esempio

La seguente query mostra i dettagli della query scaricata, inclusi formato, righe e numero di file del comando unload.

```
SELECT query_id,
       file_format,
       start_time,
       duration,
       unloaded_rows,
       unloaded_files_count
FROM sys_unload_history
ORDER BY query_id,
file_format limit 100;
```

Output di esempio.

```
 query_id | file_format |         start_time         | duration | unloaded_rows |
 unloaded_files_count
```

```
----------+-------------+----------------------------+----------+--------------
+----------------------
   527067 | Text        | 2022-02-09 05:18:35.844452 |  5932478 |           10 |
            1
```

SYS_USERLOG

Registra i dettagli per le seguenti modifiche a un utente di database:

- Create user (Crea utente)
- Rimozione dell'utente
- Modifica di un utente (assegnazione di un nuovo nome)
- Modifica di un utente (modifica delle proprietà)

È possibile eseguire una query su questa visualizzazione per visualizzare informazioni sui gruppi di lavoro serverless e sui cluster predisposti.

SYS_USERLOG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'identificatore dell'utente che ha inviato lo scaricamento.
user_name	character(50)	Nome utente dell'utente interessato dalla modifica.
original_user_name	character(50)	Il nome utente originale in un'operazione di assegnazi one di un nuovo nome. Per tutte le altre operazioni, questo campo è vuoto.

Nome colonna	Tipo di dati	Descrizione
action	character(10)	Operazione che si è verificata. I valori validi sono alter, create, drop e rename.
has_create_db_privs	integer	Se true (valore pari a 1), indica che l'utente ha creato delle autorizzazioni del database.
is_superuser	integer	Se true (un valore pari a 1), l'utente può aggiornare i cataloghi di sistema.
has_update_catalog_privs	integer	Se true (un valore pari a 1), l'utente può aggiornare i cataloghi di sistema.
password_expiration	timestamp	Data di scadenza della password.
session_id	integer	L'ID di processo.
transaction_id	bigint	L'ID transazione.
record_time	timestamp	Ora in UTC in cui è stata avviata la query.

Query di esempio

Il seguente esempio esegue quattro operazioni da parte dell'utente, poi interroga la vista SYS_USERLOG.

```
CREATE USER userlog1 password 'Userlog1';
ALTER USER userlog1 createdb createuser;
ALTER USER userlog1 rename  to userlog2;
DROP user userlog2;
```

```
SELECT user_id, user_name, original_user_name, action, has_create_db_privs,
 is_superuser from SYS_USERLOG order by record_time desc;
```

```
user_id |  user_name | original_user_name |  action | has_create_db_privs |
 is_superuser
--------+------------+--------------------+---------+---------------------+------------
    108 |  userlog2  |                    | drop    |                   1 |   1
    108 |  userlog2  |     userlog1       | rename  |                   1 |   1
    108 |  userlog1  |                    | alter   |                   1 |   1
    108 |  userlog1  |                    | create  |                   0 |   0
 (4 rows)
```

SYS_VACUUM_HISTORY

Utilizzare SYS_VACUUM_HISTORY per visualizzare i dettagli delle query di vacuum. Per ulteriori informazioni sul comando VACUUM, consulta VACUUM.

SYS_VACUUM_HISTORY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
user_id	integer	L'ID dell'utente che ha avviato la query.
transaction_id	Long	ID di transazione per l'istruzione VACUUM.
query_id	Long	L'identificatore della query per l'istruzione VACUUM. Puoi collegare questa tabella alla vista STL_QUERY_DETAIL per visualizzare le singole istruzioni SQL eseguite per

Nome colonna	Tipo di dati	Descrizione
		una determinata transazione VACUUM. Se sottoponi a vacuum l'intero database, ogni tabella è sottoposta a vacuum in una transazione separata. Per le operazioni automatiche VACUUM, questo valore è nullo.
database_name	text	Nome del database.
schema_name	text	Il nome dello schema.
table_name	text	Nome della tabella.
table_id	integer	L'ID della tabella.
vacuum_type	carattere	Il tipo di operazione VACUUM. I valori possibili sono i seguenti: • **`Delete`** • **`Sort`** • **`Reindex`** • **`Recluster`** • **`Full`** Per ulteriori informazioni sui tipi di vacuum, consultare VACUUM.
is_automatic	booleano	`true` se l'operazione è un vacuum automatico. In caso contrario, `false`.

Nome colonna	Tipo di dati	Descrizione
status	carattere	Descrizione dell'attività attualmente svolta come parte dell'operazione di vacuum: • Inizializzazione • Ordina • Unione • Eliminazione • Select • Non riuscito • Completa • Saltato • Creazione dell'ordine INTERLEAVED SORTKEY
start_time	timestamp	L'ora in cui è iniziata l'operazione vacuum.
end_time	timestamp	L'ora in cui è iniziata l'operazione vacuum. Se l'operazione è in corso, questo campo è vuoto.
record_time	timestamp	L'ora in cui l'operazione di aspirazione è stata registrata in SYS_VACUUM_HISTORY.
durata	integer	Il numero di microsecondi tra l'inizio e la fine dell'operazione di vacuum. Se l'operazione di vacuum è in corso, questo campo è vuoto.

Nome colonna	Tipo di dati	Descrizione
rows_before_vacuum	bigint	L'effettivo numero di righe nella tabella più le righe eliminate ancora archiviate su disco (in attesa di essere sottoposte a vacuum).
size_before_vacuum	integer	La dimensione della tabella prima dell'inizio dell'oper azione di vacuum, in MB.
reclaimable_rows	bigint	Il numero di file stimato dall'operazione di vacuum che verranno recuperate prima di iniziare.
reclaimed_rows	bigint	Il numero di righe recuperate dall'operazione di vacuum.
reclaimed_blocks	bigint	Il numero di righe recuperati dall'operazione di vacuum.
sortedrows_before_vacuum	integer	Il numero di righe ordinate nella tabella prima dell'avvio dell'operazione di vacuum.
sortedrows_after_vacuum	integer	Il numero aggiuntivo di righe ordinate nella tabella dopo il termine dell'operazione di vacuum. Questo non include le righe contate in `sortedrows_before_vacuum` .

Mappatura delle viste di sistema per la migrazione alle viste di monitoraggio SYS

Quando esegui la migrazione di un cluster con provisioning da Amazon Redshift ad Amazon Redshift serverless, le query di monitoraggio o diagnostica potrebbero fare riferimento a viste di sistema disponibili solo nei cluster con provisioning. Puoi aggiornare le query per utilizzare le viste di monitoraggio SYS. Questa pagina fornisce mappature di visualizzazione solo per SYS, a cui potete fare riferimento durante l'aggiornamento delle interrogazioni.

Argomenti

- SYS_ANALYZE_COMPRESSION_HISTORY
- SYS_MV_REFRESH_HISTORY
- SYS_MV_STATE
- SYS_PROCEDURE_CALL
- SYS_PROCEDURE_MESSAGES
- SYS_UDF_LOG
- SYS_USERLOG
- SYS_SCHEMA_QUOTA_VIOLATIONS
- SYS_SPATIAL_SIMPLIFY

SYS_QUERY_HISTORY

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_QUERY_HISTORY.

- STL_DDLTEXT
- STL_ERROR
- STL_QUERY
- STL_UTILITYTEXT
- STL_WLM_QUERY
- STV_INFLIGHT
- STV_RECENTS
- STV_WLM_QUERY_STATE
- SVL_COMPILE
- SVL_MULTI_STATEMENT_VIOLATIONS
- SVL_QLOG
- SVL_QUERY_QUEUE_INFO
- SVL_STATEMENTTEXT
- SVL_TERMINATE

SYS_QUERY_DETAIL

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_QUERY_DETAIL.

- STL_AGGR
- STL_ALERT_EVENT_LOG
- STL_BCAST
- STL_DELETE
- STL_DIST
- STL_EXPLAIN
- STL_HASH
- STL_HASHJOIN
- STL_INSERT
- STL_LIMIT
- STL_MERGE
- STL_MERGEJOIN
- STL_NESTLOOP
- STL_PARSE
- STL_PLAN_INFO
- STL_PROJECT
- STL_QUERY_METRICS
- STL_RETURN
- STL_SAVE
- STL_SCAN
- STL_SORT
- STL_STREAM_SEGS
- STL_UNIQUE
- STL_WINDOW
- STV_EXEC_STATE
- STV_QUERY_METRICS
- SVCS_QUERY_SUMMARY
- SVL_QUERY_METRICS
- SVL_QUERY_METRICS_SUMMARY

- SVL_QUERY_REPORT
- SVL_QUERY_SUMMARY
- SVV_QUERY_STATE

SYS_RESTORE_LOG

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_RESTORE_LOG.

- SVL_RESTORE_ALTER_TABLE_PROGRESS

SYS_RESTORE_STATE

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_RESTORE_STATE.

- STV_XRESTORE_ALTER_QUEUE_STATE

SYS_TRANSACTION_HISTORY

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_TRANSACTION_HISTORY.

- STL_COMMIT_STATS
- STL_TR_CONFLICT
- STL_UNDONE

SYS_QUERY_TEXT

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_QUERY_TEXT.

- STL_QUERYTEXT

SYS_CONNECTION_LOG

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_CONNECTION_LOG.

- STL_CONNECTION_LOG

SYS_SESSION_HISTORY

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_SESSION_HISTORY.

- STL_SESSIONS
- STL_RESTARTED_SESSIONS
- STV_SESSIONS

SYS_LOAD_DETAIL

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_LOAD_DETAIL.

- STL_LOAD_COMMITS

SYS_LOAD_HISTORY

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_LOAD_HISTORY.

- STL_LOAD_COMMITS

SYS_LOAD_ERROR_DETAIL

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_LOAD_ERROR_DETAIL.

- STL_LOADERROR_DETAIL
- STL_LOAD_ERRORS

SYS_UNLOAD_HISTORY

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_UNLOAD_HISTORY.

- STL_UNLOAD_LOG

SYS_UNLOAD_DETAIL

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_UNLOAD_DETAIL.

- STL_UNLOAD_LOG

SYS_COPY_REPLACEMENTS

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_COPY_REPLACEMENTS.

- STL_REPLACEMENTS

SYS_DATASHARE_USAGE_CONSUMER

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_DATASHARE_USAGE_CONSUMER.

- SVL_DATASHARE_USAGE_CONSUMER

SYS_DATASHARE_USAGE_PRODUCER

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_DATASHARE_USAGE_PRODUCER.

- SVL_DATASHARE_USAGE_PRODUCER

SYS_DATASHARE_CROSS_REGION_USAGE

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_DATASHARE_CROSS_REGION_USAGE.

- SVL_DATASHARE_CROSS_REGION_USAGE

SYS_DATASHARE_CHANGE_LOG

Alcune o tutte le colonne della tabella seguente sono definite anche in SYS_DATASHARE_CHANGE_LOG.

- SVL_DATASHARE_CHANGE_LOG

SYS_EXTERNAL_QUERY_DETAIL

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_EXTERNAL_QUERY_DETAIL.

- SVL_FEDERATED_QUERY
- SVL_S3LIST
- SVL_S3QUERY
- SVL_S3QUERY_SUMMARY

SYS_EXTERNAL_QUERY_ERROR

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_EXTERNAL_QUERY_ERROR.

- SVL_SPECTRUM_SCAN_ERROR

SYS_VACUUM_HISTORY

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_VACUUM_HISTORY.

- STL_VACUUM
- SVL_VACUUM_PERCENTAGE
- SVV_VACUUM_PROGRESS
- SVV_VACUUM_SUMMARY

SYS_ANALYZE_HISTORY

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_ANALYZE_HISTORY.

- STL_ANALYZE

SYS_ANALYZE_COMPRESSION_HISTORY

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_ANALYZE_COMPRESSION_HISTORY.

- STL_ANALYZE_COMPRESSION

SYS_MV_REFRESH_HISTORY

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_MV_REFRESH_HISTORY.

- SVL_MV_REFRESH_STATUS

SYS_MV_STATE

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_MV_STATE.

- STL_MV_STATE

SYS_PROCEDURE_CALL

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_PROCEDURE_CALL.

- SVL_STORED_PROC_CALL

SYS_PROCEDURE_MESSAGES

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_PROCEDURE_MESSAGES.

- SVL_STORED_PROC_MESSAGES

SYS_UDF_LOG

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_UDF_LOG.

- SVL_UDF_LOG

SYS_USERLOG

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_USERLOG.

- STL_USERLOG

SYS_SCHEMA_QUOTA_VIOLATIONS

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_SCHEMA_QUOTA_VIOLATIONS.

- STL_SCHEMA_QUOTA_VIOLATIONS

SYS_SPATIAL_SIMPLIFY

Alcune o tutte le colonne delle tabelle seguenti sono definite anche in SYS_SPATIAL_SIMPLIFY.

- SVL_SPATIAL_SIMPLIFY

Monitoraggio del sistema (solo con provisioning)

È possibile eseguire query sulle seguenti tabelle e viste di sistema sui cluster con provisioning. Le viste e le tabelle di sistema STL e STV contengono un sottoinsieme di dati presenti in diverse tabelle di sistema. Queste forniscono un accesso più rapido e semplice ai dati di query comuni presenti in tali tabelle.

Le viste SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento simultaneo. Le viste SVL forniscono informazioni solo per le query eseguite sul cluster principale, ad eccezione di SVL_STATEMENTTEXT. SVL_STATEMENTTEXT può contenere informazioni per le query eseguite su cluster di scalabilità simultanea e sul cluster principale.

Argomenti

- Viste STL per la registrazione
- Tabelle STV per dati di snapshot
- Visualizzazioni SVCS per i cluster principale e con scalabilità simultanea
- Viste SVL per il cluster principale

Viste STL per la registrazione

Le viste di sistema STL sono generate dai file di log di Amazon Redshift per fornire una cronologia del sistema.

Questi file si trovano in ogni nodo nel cluster del data warehouse. Le viste STL prendono le informazioni dai log e le formattano in tabelle utilizzabili da amministratori di sistema.

Conservazione dei log: le viste di sistema STL mantengono per sette giorni la cronologia dei log. La conservazione dei log è garantita per tutte le dimensioni e i tipi di nodi di cluster e non è interessata dalle modifiche del carico di lavoro del cluster. La conservazione dei log inoltre non è interessata dallo stato del cluster, ad esempio se il cluster è in pausa. La cronologia dei log è inferiore a sette giorni solo nel caso in cui il cluster è nuovo. Non è necessario intraprendere alcuna azione per conservare i log, ma è necessario copiare periodicamente i dati di log su altre tabelle o scaricarli su Amazon S3 per conservare i dati di log che risalgono a più di 7 giorni fa.

Argomenti

- STL_LIMIT
- STL_LOAD_COMMITS
- STL_LOAD_ERRORS
- STL_LOADERROR_DETAIL
- STL_MERGE
- STL_MERGEJOIN
- STL_MV_STATE
- STL_NESTLOOP
- STL_PARSE
- STL_PLAN_INFO
- STL_PROJECT
- STL_QUERY
- STL_QUERY_METRICS
- STL_QUERYTEXT
- STL_REPLACEMENTS
- STL_RESTARTED_SESSIONS
- STL_RETURN
- STL_S3CLIENT
- STL_S3CLIENT_ERROR
- STL_SAVE
- STL_SCAN
- STL_SCHEMA_QUOTA_VIOLATIONS
- STL_SESSIONS
- STL_SORT
- STL_SSHCLIENT_ERROR
- STL_STREAM_SEGS
- STL_TR_CONFLICT
- STL_UNDONE
- STL_UNIQUE
- STL_UNLOAD_LOG

- STL_USAGE_CONTROL
- STL_USERLOG
- STL_UTILITYTEXT
- STL_VACUUM
- STL_WINDOW
- STL_WLM_ERROR
- STL_WLM_RULE_ACTION
- STL_WLM_QUERY

STL_AGGR

Analizza le fasi di esecuzione aggregate per le query. Queste fasi si verificano durante l'esecuzione di funzioni aggregate e clausole GROUP BY.

STL_AGGR è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_AGGR contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.

Nome colonna	Tipo di dati	Descrizione
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
byte	bigint	Dimensione, in byte, di tutte le righe di output della fase.
slots	integer	Numero di bucket di hash.
occupied	integer	Numero di slot che contengono record.
maxlength	integer	Dimensioni dello slot più grande.
tbl	integer	ID tabella.
is_diskbased	character(1)	Se true (t), la query è stata eseguita come un'operazione basata su disco. Se false (f), la query è stata eseguita in memoria.
workmem	bigint	Numero di byte della memoria di lavoro assegnati alla fase.

Nome colonna	Tipo di dati	Descrizione
tipo	character(6)	Il tipo di fase. I valori validi sono: • HASHED. Indica che la fase ha utilizzato un'aggregazione non ordinata, raggruppata. • PLAIN. Indica che la fase ha utilizzato un'aggregazione non raggruppata, scalare. • SORTED. Indica che la fase ha utilizzato un'aggregazione ordinata, raggruppata.
resizes	integer	Queste informazioni sono solo per uso interno.
flushable	integer	Queste informazioni sono solo per uso interno.

Query di esempio

Restituisce informazioni sulle fasi di esecuzione di aggregazione per SLICE 1 e TBL 239.

```
select query, segment, bytes, slots, occupied, maxlength, is_diskbased, workmem, type
from stl_aggr where slice=1 and tbl=239
order by rows
limit 10;
```

```
 query | segment | bytes |  slots  | occupied | maxlength | is_diskbased |  workmem  |
 type
-------+---------+-------+---------+----------+-----------+--------------+-----------
+--------
   562 |       1 |     0 | 4194304 |        0 |         0 | f            | 383385600 |
 HASHED
   616 |       1 |     0 | 4194304 |        0 |         0 | f            | 383385600 |
 HASHED
   546 |       1 |     0 | 4194304 |        0 |         0 | f            | 383385600 |
 HASHED
   547 |       0 |     8 |       0 |        0 |         0 | f            |         0 |
 PLAIN
   685 |       1 |    32 | 4194304 |        1 |         0 | f            | 383385600 |
 HASHED
```

```
  652 |       0 |     8 |       0 |        0 |         0 | f             |         0 |
 PLAIN
  680 |       0 |     8 |       0 |        0 |         0 | f             |         0 |
 PLAIN
  658 |       0 |     8 |       0 |        0 |         0 | f             |         0 |
 PLAIN
  686 |       0 |     8 |       0 |        0 |         0 | f             |         0 |
 PLAIN
  695 |       1 |    32 | 4194304 |        1 |         0 | f             | 383385600 |
 HASHED
(10 rows)
```

STL_ALERT_EVENT_LOG

Registra un avviso quando il query optimizer identifica delle condizioni che potrebbero indicare problemi di prestazioni. Utilizza la vista STL_ALERT_EVENT_LOG per identificare le possibilità di miglioramento delle performance della query.

Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. Per ulteriori informazioni, consulta Elaborazione query.

STL_ALERT_EVENT_LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_ALERT_EVENT_LOG contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.

Nome colonna	Tipo di dati	Descrizione
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
pid	integer	ID di processo associato all'istruzione e alla sezione. La stessa query potrebbe avere più PID se viene eseguita su più sezioni.
xid	bigint	ID di transazione associato all'istruzione.
evento	character (1024)	Descrizione dell'evento di avviso.
solution	character (1024)	Soluzione consigliata.
event_time	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .

Note per l'utilizzo

È possibile utilizzare lo STL_ALERT_EVENT_LOG per identificare potenziali problemi nelle query, poi, per ottimizzare la progettazione del database e riscrivere le query, segui le procedure in Ottimizzazione delle prestazioni delle query. STL_ALERT_EVENT_LOG registra i seguenti avvisi:

- Statistiche mancanti

  Mancano le statistiche. Esegui ANALYZE in seguito a caricamenti di dati o aggiornamenti significativi e utilizza STATUPDATE con le operazioni COPY. Per ulteriori informazioni, consulta Best practice di Amazon Redshift per la progettazione di query.

- Loop nidificato

  Un loop nidificato è solitamente un prodotto cartesiano. Valuta la query per garantire che tutte le tabelle che partecipano siano unite in modo efficiente.

- Filtro molto selettivo

  Il rapporto tra righe restituite e righe scansionate è minore di 0,05. Le righe sottoposte a scansione corrispondono al valore di `rows_pre_user_filter` e le righe restituite sono il valore delle righe nella vista di sistema STL_SCAN. Indica che la query sta eseguendo la scansione di un numero insolitamente grande di righe per determinare il set di risultati. Questo può essere dovuto a chiavi di ordinamento mancanti o non corrette. Per ulteriori informazioni, consulta Utilizzo delle chiavi di ordinamento.

- Righe fantasma eccessive

  Una scansione ha ignorato un numero piuttosto grande di righe contrassegnate come cancellate ma non sottoposte a vacuum o righe che sono state inserite ma non eseguite. Per ulteriori informazioni, consulta Vacuum delle tabelle.

- Distribuzione estesa

  Più di 1.000.000 di righe sono state redistribuite per un hash join o un'aggregazione. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati.

- Trasmissione estesa

  Più di 1.000.000 di righe sono state trasmesse per un hash join. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati.

- Esecuzione seriale

  Uno stile di redistribuzione DS_DIST_ALL_INNER è stato indicato nel piano di query, il che forza un'esecuzione seriale perché l'intera tabella interna è stata redistribuita in un solo nodo. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati.

Query di esempio

La query seguente mostra eventi di avviso per quattro query.

```
SELECT query, substring(event,0,25) as event,
substring(solution,0,25) as solution,
trim(event_time) as event_time from stl_alert_event_log order by query;
```

```
 query |             event             |          solution            |     event_time

-------+-------------------------------+------------------------------
+---------------------
  6567 | Missing query planner statist | Run the ANALYZE command      | 2014-01-03
 18:20:58
  7450 | Scanned a large number of del | Run the VACUUM command to rec| 2014-01-03
 21:19:31
  8406 | Nested Loop Join in the query | Review the join predicates to| 2014-01-04
 00:34:22
 29512 | Very selective query filter:r | Review the choice of sort key| 2014-01-06
 22:00:00

(4 rows)
```

STL_ANALYZE

Registra i dettagli per le operazioni ANALYZE.

STL_ANALYZE è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_ANALYZE_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato la voce.
xid	Long	L'ID transazione.
database	char(30)	Nome del database.
table_id	integer	L'ID della tabella.

Nome colonna	Tipo di dati	Descrizione
status	char(15)	Risultato del comando di analisi. I valori possibili sono `Full`, `Skipped` e `PredicateColumn` .
righe	double	Il numero totale di righe nella tabella.
modified_rows	double	Il numero complessivo di righe modificate dopo l'ultima operazione ANALYZE.
threshold_percent	integer	Il valore del parametro `analyze_threshold_percent` .
is_auto	char(1)	Il valore è true (`t`) se l'operazione includeva per impostazione predefinita un'operazione di analisi di Amazon Redshift. Il valore è false (`f`) se il comando ANALYZE è stato eseguito in modo esplicito.
starttime	timestamp	L'ora in UTC in cui l'operazione di analisi ha iniziato l'esecuzione.
endtime	timestamp	L'ora in UTC in cui l'operazione di analisi ha terminato l'esecuzione.
prevtime	timestamp	L'ora in UTC in cui la tabella è stata analizzata in precedenza.
num_predicate_cols	integer	Il numero attuale di colonne di predicato nella tabella.
num_new_predicate_cols	integer	Il numero delle nuove colonne di predicato nella tabella dall'operazione di analisi precedente.
is_background	character(1)	Il valore è true (`t`) se l'analisi è stata eseguita da un'operazione di analisi automatica. Altrimenti, il valore predefinito è false (`f`).

Nome colonna	Tipo di dati	Descrizione
auto_analyze_phase	character(100)	Riservato per uso interno.
schema_name	char(128)	Il nome dello schema per la tabella.
table_name	char(136)	Nome della tabella.

Query di esempio

L'esempio seguente collega STV_TBL_PERM per mostrare il nome della tabella e i dettagli di esecuzione.

```
select distinct a.xid, trim(t.name) as name, a.status, a.rows, a.modified_rows,
 a.starttime, a.endtime
from stl_analyze a
join stv_tbl_perm t  on t.id=a.table_id
where name = 'users'
order by starttime;

xid    | name  | status          | rows  | modified_rows | starttime           |
 endtime
-------+-------+-----------------+-------+---------------+---------------------
+---------------------
  1582 | users | Full            | 49990 |         49990 | 2016-09-22 22:02:23 |
 2016-09-22 22:02:28
244287 | users | Full            | 24992 |         74988 | 2016-10-04 22:50:58 |
 2016-10-04 22:51:01
244712 | users | Full            | 49984 |         24992 | 2016-10-04 22:56:07 |
 2016-10-04 22:56:07
245071 | users | Skipped         | 49984 |             0 | 2016-10-04 22:58:17 |
 2016-10-04 22:58:17
245439 | users | Skipped         | 49984 |          1982 | 2016-10-04 23:00:13 |
 2016-10-04 23:00:13
(5 rows)
```

STL_ANALYZE_COMPRESSION

Registra i dettagli per le operazioni di analisi della compressione durante i comandi COPY o ANALYZE COMPRESSION.

STL_ANALYZE_COMPRESSION è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_ANALYZE_COMPRESSION_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato la voce.
start_time	timestamp	L'ora di inizio dell'operazione di analisi della compressione.
xid	bigint	L'ID transazione dell'operazione di analisi della compressione.
tbl	integer	L'ID della tabella che è stata analizzata.
tablename	character(128)	Il nome della tabella che è stata analizzata.
col	integer	L'indice della colonna nella tabella che è stata analizzata per determinare la codifica della compressione.
old_encoding	character(15)	Il tipo di codifica prima dell'analisi della compressione.
new_encoding	character(15)	Il tipo di codifica dopo l'analisi della compressione.
mode	character(14)	I valori possibili sono:

Nome colonna	Tipo di dati	Descrizione
		PRESET Specifica che `new_encoding` è determinato dal comando Amazon Redshift COPY sulla base del tipo di dati della colonna. Nessun dato viene campionato. ATTIVATO Specifica che `new_encoding` è determinato dal comando Amazon Redshift COPY sulla base di un'analisi dei dati di esempio. ANALYZE ONLY Specifica che `new_encoding` è determinato dal comando Amazon Redshift ANALYZE COMPRESSION sulla base di un'analisi dei dati di esempio. Tuttavia, il tipo di codifica della colonna analizzata non viene modificato.
best_compression_encoding	character(15)	Il tipo di codifica che offre il miglior rapporto di compressione.
byte_raccomandati	character(15)	I byte utilizzati adottando la nuova codifica.
best_compression_bytes	character(15)	I byte utilizzati adottando la migliore codifica di compressione.
ndv	bigint	Il numero di valori distinti nelle righe campionate.

Query di esempio

Nell'esempio seguente vengono esaminati i dettagli dell'analisi della compressione sulla tabella `lineitem` dall'ultimo comando COPY eseguito nella stessa sessione.

```
select xid, tbl, btrim(tablename) as tablename, col, old_encoding, new_encoding,
 best_compression_encoding, mode
from stl_analyze_compression
where xid = (select xid from stl_query where query = pg_last_copy_id()) order by col;

 xid  |  tbl   | tablename | col |  old_encoding   |  new_encoding   |
 best_compression_encoding |      mode
------+--------+-----------+-----+-----------------+-----------------
+---------------------------+----------------
 5308 | 158961 | $lineitem |   0 | mostly32        | az64            | delta
          | ON
 5308 | 158961 | $lineitem |   1 | mostly32        | az64            | az64
          | ON
 5308 | 158961 | $lineitem |   2 | lzo             | az64            | az64
          | ON
 5308 | 158961 | $lineitem |   3 | delta           | az64            | az64
          | ON
 5308 | 158961 | $lineitem |   4 | bytedict        | az64            | bytedict
          | ON
 5308 | 158961 | $lineitem |   5 | mostly32        | az64            | az64
          | ON
 5308 | 158961 | $lineitem |   6 | delta           | az64            | az64
          | ON
 5308 | 158961 | $lineitem |   7 | delta           | az64            | az64
          | ON
 5308 | 158961 | $lineitem |   8 | lzo             | lzo             | lzo
          | ON
 5308 | 158961 | $lineitem |   9 | runlength       | runlength       | runlength
          | ON
 5308 | 158961 | $lineitem |  10 | delta           | az64            | az64
          | ON
 5308 | 158961 | $lineitem |  11 | delta           | az64            | az64
          | ON
 5308 | 158961 | $lineitem |  12 | delta           | az64            | az64
          | ON
 5308 | 158961 | $lineitem |  13 | bytedict        | bytedict        | bytedict
          | ON
 5308 | 158961 | $lineitem |  14 | bytedict        | bytedict        | bytedict
          | ON
 5308 | 158961 | $lineitem |  15 | text255         | text255         | text255
          | ON
(16 rows)
```

STL_BCAST

Registra le informazioni sull'attività di rete durante l'esecuzione delle fasi di query che trasmettono dati. Il traffico di rete è acquisito dal numero di righe, dai byte e dai pacchetti inviati nella rete durante una data fase in una data sezione. La durata della fase è la differenza tra l'ora di inizio e l'ora di fine registrazione.

Per identificare le fasi di trasmissione in una query, cerca le etichette bcast nella visualizzazione SVL_QUERY_SUMMARY o esegui il comando EXPLAIN e cerca gli attributi di fase che includono bcast.

STL_BCAST è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_BCAST contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.

Nome colonna	Tipo di dati	Descrizione
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
byte	bigint	Dimensione, in byte, di tutte le righe di output della fase.
packets	integer	Numero complessivo di pacchetti inviati nella rete.

Query di esempio

L'esempio seguente restituisce le informazioni di trasmissione per le query in cui ci sono uno o più pacchetti e la differenza tra l'inizio e la fine della query è di un secondo o più.

```
select query, slice, step, rows, bytes, packets, datediff(seconds, starttime, endtime)
from stl_bcast
where packets>0 and datediff(seconds, starttime, endtime)>0;
```

```
 query | slice | step | rows | bytes | packets | date_diff
-------+-------+------+------+-------+---------+-----------
   453 |     2 |    5 |    1 |   264 |       1 |         1
   798 |     2 |    5 |    1 |   264 |       1 |         1
  1408 |     2 |    5 |    1 |   264 |       1 |         1
  2993 |     0 |    5 |    1 |   264 |       1 |         1
```

```
  5045 |     3 |    5 |    1 |   264 |       1 |         1
  8073 |     3 |    5 |    1 |   264 |       1 |         1
  8163 |     3 |    5 |    1 |   264 |       1 |         1
  9212 |     1 |    5 |    1 |   264 |       1 |         1
  9873 |     1 |    5 |    1 |   264 |       1 |         1
(9 rows)
```

STL_COMMIT_STATS

Fornisce parametri correlati alle prestazioni di commit, inclusi la tempistica delle varie fasi di commit e il numero di blocchi di commit. Esegui una query su STL_COMMIT_STATS per determinare quale parte di una transazione è stata dedicata al commit e l'entità della messa in coda.

STL_COMMIT_STATS è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_TRANSACTION_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
xid	bigint	Id di transazione in corso di commit.
node	integer	Numero di nodi. -1 è il nodo principale.
startqueue	timestamp	Inizio della messa in coda per il commit.
startwork	timestamp	Inizio del commit.
endflush	timestamp	Fine della fase di scarico dei blocchi dirty.
endstage	timestamp	Fine della fase di gestione temporanea di metadati.
endlocal	timestamp	Fine della fase di commit locale.
startglobal	timestamp	Inizio della fase globale.

Nome colonna	Tipo di dati	Descrizione
endtime	timestamp	Fine del commit.
queuelen	bigint	Numero di transazioni precedenti a questa transazione nella coda di commit.
permblocks	bigint	Numero di blocchi permanenti esistenti al momento di questo commit.
newblocks	bigint	Numero di nuovi blocchi permanenti al momento di questo commit.
dirtyblocks	bigint	Numero di blocchi scritti come parte di questo commit.
headers	bigint	Numero delle intestazioni dei blocchi scritte come parte di questo commit.
numxids	integer	Il numero di transazioni DML attive.
oldestxid	bigint	Lo XID della più vecchia transazione DML attiva.
extwritelatency	bigint	Queste informazioni sono solo per uso interno.
metadatawritten	int	Queste informazioni sono solo per uso interno.
tombstonedblocks	bigint	Queste informazioni sono solo per uso interno.
tossedblocks	bigint	Queste informazioni sono solo per uso interno.
batched_by	bigint	Queste informazioni sono solo per uso interno.

Query di esempio

```
select node, datediff(ms,startqueue,startwork) as queue_time,
```

```
datediff(ms, startwork, endtime) as commit_time, queuelen
from stl_commit_stats
where xid = 2574
order by node;

node | queue_time   | commit_time | queuelen
-----+--------------+-------------+---------
  -1 |            0 |         617 |        0
   0 | 444950725641 |         616 |        0
   1 | 444950725636 |         616 |        0
```

STL_CONNECTION_LOG

Registra i tentativi di autenticazione, insieme alle connessioni.

STL_CONNECTION_LOG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_CONNECTION_LOG. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
evento	character(50)	Evento di connessione o autenticazione.
recordtime	timestamp	Ora in cui l'evento si è verificato.
remotehost	character(45)	Nome o indirizzo IP dell'host remoto.
remoteport	character(32)	Numero di porta per l'host remoto.
pid	integer	ID di processo associato all'istruzione.
dbname	character(50)	Nome del database.
username	character(50)	Nome dell'utente.

Nome colonna	Tipo di dati	Descrizione
authmethod	character(32)	Metodo di autenticazione.
durata	integer	Durata di connessione in microsecondi.
sslversion	character(50)	Versione Secure Sockets Layer (SSL).
sslcipher	character(128)	Crittografia SSL.
mtu	integer	Unità di trasmissione massima (MTU).
sslcompression	character(64)	Tipo di compressione SSL.
sslexpansion	character(64)	Tipo di espansione SSL.
iamauthguid	character(36)	L'ID di autenticazione IAM per la CloudTrail richiesta.
application_name	character(250)	Il nome iniziale o aggiornato dell'applicazione per una sessione.
os_version	character(64)	La versione del sistema operativo presente sul computer client che si connette al cluster Amazon Redshift.
driver_version	character(64)	La versione del driver ODBC o JDBC che si connette al tuo cluster Amazon Redshift dai tuoi strumenti client SQL di terze parti.
plugin_name	character(32)	Il nome del plug-in utilizzato per connettersi al cluster Amazon Redshift.

Nome colonna	Tipo di dati	Descrizione
protocol_ version	integer	La versione del protocollo interno utilizzato dal driver Amazon Redshift per stabilire la connessione con il server. Le versioni del protocollo sono negoziate tra driver e server. La versione descrive le funzionalità disponibili. I valori validi includono: • 0 (BASE_SERVER_PROTOCOL_VERSION) • 1 (EXTENDED_RESULT_METADATA_SERVER_PRO TOCOL_VERSION): per salvare un'andata e ritorno per query, il server invia ulteriori informazioni sui metadati del set di risultati. • 2 (BINARY_PROTOCOL_VERSION): a seconda del tipo di dati del set di risultati, il server invia dati in formato binario. • 3 (EXTENDED2_RESULT_METADATA_SERVER_PR OTOCOL_VERSION): il server invia informazioni di distinzione tra maiuscole e minuscole (regole di confronto) di una colonna.
sessionid	character(36)	L'identificatore univoco globale per la sessione attuale. L'ID di sessione persiste con il riavvio dell'errore del nodo.
compressi one	character(16)	L'algoritmo di compressione utilizzato per la connessione.

Query di esempio

Per visualizzare i dettagli per le connessioni aperte, esegui la seguente query.

```
select recordtime, username, dbname, remotehost, remoteport
from stl_connection_log
where event = 'initiating session'
and pid not in
(select pid from stl_connection_log
where event = 'disconnecting session')
order by 1 desc;
```

```
recordtime          | username    | dbname     | remotehost    | remoteport

--------------------+-------------+------------+---------------+------------
2014-11-06 20:30:06 | rdsdb       | dev        | [local]       |

2014-11-06 20:29:37 | test001     | test       | 10.49.42.138  | 11111

2014-11-05 20:30:29 | rdsdb       | dev        | 10.49.42.138  | 33333

2014-11-05 20:28:35 | rdsdb       | dev        | [local]       |
(4 rows)
```

L'esempio seguente riflette un tentativo di autenticazione non riuscito e una connessione e una disconnessione andate a buon fine.

```
select event, recordtime, remotehost, username
from stl_connection_log order by recordtime;

            event      |         recordtime         |  remotehost   | username

-----------------------+----------------------------+---------------+---------
authentication failure | 2012-10-25 14:41:56.96391 | 10.49.42.138 | john

authenticated          | 2012-10-25 14:42:10.87613 | 10.49.42.138 | john

initiating session     | 2012-10-25 14:42:10.87638 | 10.49.42.138 | john

disconnecting session  | 2012-10-25 14:42:19.95992 | 10.49.42.138 | john

(4 rows)
```

L'esempio seguente mostra la versione del driver ODBC, il sistema operativo sul computer client e il plug-in utilizzato per connettersi al cluster Amazon Redshift. In questo esempio, il plug-in utilizzato è per l'autenticazione standard del driver ODBC utilizzando un nome di accesso e una password.

```
select driver_version, os_version, plugin_name from stl_connection_log;

driver_version                          |  os_version                        |
 plugin_name
-----------------------------------------+------------------------------------
+--------------------
Amazon Redshift ODBC Driver 1.4.15.0001 | Darwin 18.7.0 x86_64               | none
```

```
Amazon Redshift ODBC Driver 1.4.15.0001 | Linux 4.15.0-101-generic x86_64   | none
```

Nell'esempio seguente viene illustrata la versione del sistema operativo sul computer client, la versione del driver e la versione del protocollo.

```
select os_version, driver_version, protocol_version from stl_connection_log;

os_version                     |  driver_version              | protocol_version
-------------------------------+------------------------------+-----------------
Linux 4.15.0-101-generic x86_64 | Redshift JDBC Driver 2.0.0.0 | 2
Linux 4.15.0-101-generic x86_64 | Redshift JDBC Driver 2.0.0.0 | 2
Linux 4.15.0-101-generic x86_64 | Redshift JDBC Driver 2.0.0.0 | 2
```

STL_DDLTEXT

Acquisisce le seguenti istruzioni DDL eseguite nel sistema.

Queste istruzioni DDL includono le query e gli oggetti seguenti:

- CREATE SCHEMA, TABLE, VIEW
- DROP SCHEMA, TABLE, VIEW
- ALTER SCHEMA, TABLE

Consulta anche STL_QUERYTEXT, STL_UTILITYTEXT e SVL_STATEMENTTEXT. Queste viste forniscono una sequenza temporale dei comandi SQL eseguiti nel sistema; la cronologia è utile per la risoluzione dei problemi e per creare un percorso di verifica di tutte le attività di sistema.

Utilizza le colonne STARTTIME ed ENDTIME per scoprire quali istruzioni sono state registrate in un dato periodo. Blocchi lunghi di testo SQL sono suddivisi in righe di 200 caratteri; la colonna SEQUENCE identifica i frammenti di testo che appartengono a ogni singola istruzione.

STL_DDLTEXT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
xid	bigint	ID di transazione associato all'istruzione.
pid	integer	ID di processo associato all'istruzione.
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etiche tta definita con un comando SET QUERY GROUP. Se la query non è basata su file o il parametro QUERY_GROUP non è impostato, questo campo è vuoto.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
sequenza	integer	Quando una singola istruzione contiene più di 200 caratteri , vengono registrate delle righe aggiuntive per tale istruzion e. La sequenza 0 è la prima riga, 1 la seconda e così via.
text	character(200)	Testo SQL, in incrementi da 200 caratteri. Questo campo potrebbe contenere caratteri speciali come barra rovesciat a (\\) e nuova riga (\n).

Query di esempio

La seguente query restituisce i record che includono istruzioni DDL eseguite in precedenza.

```
select xid, starttime, sequence, substring(text,1,40) as text
```

```
from stl_ddltext order by xid desc, sequence;
```

Di seguito è riportato un output di esempio che mostra quattro istruzioni CREATE TABLE. Le istruzioni DDL vengono visualizzate nella colonna `text`, che è troncata per ragioni di leggibilità.

```
 xid  |         starttime          | sequence |                  text
------+----------------------------+----------
+----------------------------------------
 1806 | 2013-10-23 00:11:14.709851 |        0 | CREATE TABLE supplier ( s_suppkey int4
 N
 1806 | 2013-10-23 00:11:14.709851 |        1 |  s_comment varchar(101) NOT NULL )
 1805 | 2013-10-23 00:11:14.496153 |        0 | CREATE TABLE region ( r_regionkey int4
 N
 1804 | 2013-10-23 00:11:14.285986 |        0 | CREATE TABLE partsupp ( ps_partkey int8
 1803 | 2013-10-23 00:11:14.056901 |        0 | CREATE TABLE part ( p_partkey int8 NOT
 N
 1803 | 2013-10-23 00:11:14.056901 |        1 | ner char(10) NOT NULL , p_retailprice
 nu
(6 rows)
```

Ricostruzione dell'SDL archiviato

Il seguente SQL elenca le righe archiviate nella colonna `text` di STL_DDLTEXT. Le righe sono ordinate per `xid` e `sequence`. Se l'SQL originale era più lungo di 200 righe multiple di caratteri, STL_DDLTEXT può contenere più righe in `sequence`.

```
SELECT xid, sequence, LISTAGG(CASE WHEN LEN(RTRIM(text)) = 0 THEN text ELSE RTRIM(text)
 END, '') WITHIN GROUP (ORDER BY sequence) as query_statement
FROM stl_ddltext GROUP BY xid, sequence ORDER BY xid, sequence;
```

```
xid     |  sequence | query_statement
--------+-----------+----------------
7886671    0          create external schema schema_spectrum_uddh\nfrom data catalog
\ndatabase 'spectrum_db_uddh'\niam_role ''\ncreate external database if not exists;
7886752    0          CREATE EXTERNAL TABLE schema_spectrum_uddh.soccer_league\n(\n
  league_rank smallint,\n  prev_rank   smallint,\n  club_name   varchar(15),\n
 league_name varchar(20),\n  league_off  decimal(6,2),\n  le
7886752    1          ague_def  decimal(6,2),\n  league_spi  decimal(6,2),\n
 league_nspi smallint\n)\nROW FORMAT DELIMITED \n    FIELDS TERMINATED BY ',' \n
 LINES TERMINATED BY '\\n\\l'\nstored as textfile\nLOCATION 's
```

```
7886752    2          3://mybucket-spectrum-uddh/'\ntable properties
 ('skip.header.line.count'='1');
...
```

Per ricostruire l'SQL archiviato nella colonna `text` di STL_DDLTEXT, eseguire la seguente istruzione SQL. Unisce istruzioni DDL di uno o più segmenti nella colonna `text`. Prima di eseguire l'SQL ricostruito, sostituire tutti i caratteri speciali (\n) con una nuova riga nel client SQL. I risultati della seguente istruzione SELECT riuniscono tre righe in ordine di sequenza per ricostruire l'SQL, nel campo `query_statement`.

```
SELECT LISTAGG(CASE WHEN LEN(RTRIM(text)) = 0 THEN text ELSE RTRIM(text) END) WITHIN
 GROUP (ORDER BY sequence) as query_statement
FROM stl_ddltext GROUP BY xid, endtime order by xid, endtime;
```

```
query_statement
--------------
create external schema schema_spectrum_uddh\nfrom data catalog\ndatabase
 'spectrum_db_uddh'\niam_role ''\ncreate external database if not exists;
CREATE EXTERNAL TABLE schema_spectrum_uddh.soccer_league\n(\n  league_rank smallint,
\n  prev_rank   smallint,\n  club_name   varchar(15),\n  league_name varchar(20),\n
 league_off  decimal(6,2),\n  league_def  decimal(6,2),\n  league_spi  decimal(6,2),
\n  league_nspi smallint\n)\nROW FORMAT DELIMITED \n    FIELDS TERMINATED BY ',' \n
  LINES TERMINATED BY '\\n\\l'\nstored as textfile\nLOCATION 's3://mybucket-spectrum-
uddh/'\ntable properties ('skip.header.line.count'='1');
```

STL_DELETE

Analizza le fasi di esecuzione di eliminazione per le query.

STL_DELETE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_DELETE contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista

di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
tbl	integer	ID tabella.

Query di esempio

Al fine di creare una riga in STL_DELETE, l'esempio seguente inserisce una riga nella tabella EVENT e poi la elimina.

Prima, inserisci una riga nella tabella EVENT e verifica che sia stata inserita.

```
insert into event(eventid,venueid,catid,dateid,eventname)
values ((select max(eventid)+1 from event),95,9,1857,'Lollapalooza');
```

```
select * from event
where eventname='Lollapalooza'
order by eventid;
```

```
 eventid | venueid | catid | dateid |  eventname   |      starttime
---------+---------+-------+--------+--------------+---------------------
    4274 |     102 |     9 |   1965 | Lollapalooza | 2008-05-01 19:00:00
    4684 |     114 |     9 |   2105 | Lollapalooza | 2008-10-06 14:00:00
    5673 |     128 |     9 |   1973 | Lollapalooza | 2008-05-01 15:00:00
    5740 |      51 |     9 |   1933 | Lollapalooza | 2008-04-17 15:00:00
    5856 |     119 |     9 |   1831 | Lollapalooza | 2008-01-05 14:00:00
    6040 |     126 |     9 |   2145 | Lollapalooza | 2008-11-15 15:00:00
    7972 |      92 |     9 |   2026 | Lollapalooza | 2008-07-19 19:30:00
    8046 |      65 |     9 |   1840 | Lollapalooza | 2008-01-14 15:00:00
    8518 |      48 |     9 |   1904 | Lollapalooza | 2008-03-19 15:00:00
    8799 |      95 |     9 |   1857 | Lollapalooza |
(10 rows)
```

Adesso elimina la riga che hai aggiunto nella tabella EVENT e verifica che sia stata eliminata.

```
delete from event
where eventname='Lollapalooza' and eventid=(select max(eventid) from event);
```

```
select * from event
where eventname='Lollapalooza'
order by eventid;
```

```
 eventid | venueid | catid | dateid |  eventname   |      starttime
```

```
---------+---------+-------+--------+-------------+---------------------
    4274 |     102 |     9 |   1965 | Lollapalooza | 2008-05-01 19:00:00
    4684 |     114 |     9 |   2105 | Lollapalooza | 2008-10-06 14:00:00
    5673 |     128 |     9 |   1973 | Lollapalooza | 2008-05-01 15:00:00
    5740 |      51 |     9 |   1933 | Lollapalooza | 2008-04-17 15:00:00
    5856 |     119 |     9 |   1831 | Lollapalooza | 2008-01-05 14:00:00
    6040 |     126 |     9 |   2145 | Lollapalooza | 2008-11-15 15:00:00
    7972 |      92 |     9 |   2026 | Lollapalooza | 2008-07-19 19:30:00
    8046 |      65 |     9 |   1840 | Lollapalooza | 2008-01-14 15:00:00
    8518 |      48 |     9 |   1904 | Lollapalooza | 2008-03-19 15:00:00
(9 rows)
```

Poi, esegui una query su stl_delete per vedere le fasi di esecuzione dell'eliminazione. In questo esempio, la query ha restituito più di 300 righe, pertanto l'output qui sotto è stato abbreviato per ragioni di visualizzazione.

```
select query, slice, segment, step, tasknum, rows, tbl from stl_delete order by query;
```

```
 query | slice | segment | step | tasknum | rows |  tbl
-------+-------+---------+------+---------+------+--------
     7 |     0 |       0 |    1 |       0 |    0 | 100000
     7 |     1 |       0 |    1 |       0 |    0 | 100000
     8 |     0 |       0 |    1 |       2 |    0 | 100001
     8 |     1 |       0 |    1 |       2 |    0 | 100001
     9 |     0 |       0 |    1 |       4 |    0 | 100002
     9 |     1 |       0 |    1 |       4 |    0 | 100002
    10 |     0 |       0 |    1 |       6 |    0 | 100003
    10 |     1 |       0 |    1 |       6 |    0 | 100003
    11 |     0 |       0 |    1 |       8 |    0 | 100253
    11 |     1 |       0 |    1 |       8 |    0 | 100253
    12 |     0 |       0 |    1 |       0 |    0 | 100255
    12 |     1 |       0 |    1 |       0 |    0 | 100255
    13 |     0 |       0 |    1 |       2 |    0 | 100257
    13 |     1 |       0 |    1 |       2 |    0 | 100257
    14 |     0 |       0 |    1 |       4 |    0 | 100259
    14 |     1 |       0 |    1 |       4 |    0 | 100259
    ...
```

STL_DISK_FULL_DIAG

Registra le informazioni sugli errori registrati quando il disco è pieno.

STL_DISK_FULL_DIAG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
currenttime	bigint	Il giorno e l'ora in cui l'errore è stato generato in microsecondi dal 1° gennaio del 2000.
node_num	bigint	Identificatore per il nodo.
query_id	bigint	Identificatore per la query che ha causato l'errore.
temp_blocks	bigint	Il numero di blocchi temporanei creati dalla query.

Query di esempio

L'esempio seguente restituisce i dettagli sui dati archiviati in caso di errore di disco pieno.

```
select * from stl_disk_full_diag
```

L'esempio seguente converte `currenttime` in un timestamp.

```
select '2000-01-01'::timestamp + (currenttime/1000000.0)* interval '1 second' as
 currenttime,node_num,query_id,temp_blocks from pg_catalog.stl_disk_full_diag;
```

```
        currenttime         | node_num | query_id | temp_blocks
----------------------------+----------+----------+-------------
 2019-05-18 19:19:18.609338 |        0 |   569399 |       70982
 2019-05-18 19:37:44.755548 |        0 |   569580 |       70982
 2019-05-20 13:37:20.566916 |        0 |   597424 |       70869
```

STL_DIST

Registra le informazioni sull'attività di rete durante l'esecuzione delle fasi di query che distribuiscono dati. Il traffico di rete è acquisito dal numero di righe, dai byte e dai pacchetti inviati nella rete durante una data fase in una data sezione. La durata della fase è la differenza tra l'ora di inizio e l'ora di fine registrazione.

Per identificare le fasi di distribuzione in una query, cerca le etichette dist nella visualizzazione QUERY_SUMMARY o esegui il comando EXPLAIN e cerca gli attributi di fase che includono dist.

STL_DIST è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_DIST contiene solo query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.

Nome colonna	Tipo di dati	Descrizione
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
byte	bigint	Dimensione, in byte, di tutte le righe di output della fase.
packets	integer	Numero complessivo di pacchetti inviati nella rete.

Query di esempio

L'esempio seguente restituisce le informazioni di distribuzione per le query con uno o più pacchetti e durata maggiore di zero.

```
select query, slice, step, rows, bytes, packets,
datediff(seconds, starttime, endtime) as duration
from stl_dist
where packets>0 and datediff(seconds, starttime, endtime)>0
order by query
limit 10;
```

```
 query  | slice | step |  rows  |  bytes  | packets | duration
--------+-------+------+--------+---------+---------+----------
    567 |     1 |    4 |  49990 | 6249564 |     707 |        1
    630 |     0 |    5 |   8798 |  408404 |      46 |        2
```

```
    645 |     1 |    4 |   8798 |  408404 |      46 |         1
    651 |     1 |    5 | 192497 | 9226320 |    1039 |         6
    669 |     1 |    4 | 192497 | 9226320 |    1039 |         4
    675 |     1 |    5 |   3766 |  194656 |      22 |         1
    696 |     0 |    4 |   3766 |  194656 |      22 |         1
    705 |     0 |    4 |    930 |   44400 |       5 |         1
 111525 |     0 |    3 |     68 |   17408 |       2 |         1
(9 rows)
```

STL_ERROR

Registra gli errori di elaborazione interni generati dal motore di database Amazon Redshift. STL_ERROR non registra gli errori o i messaggi SQL. Le informazioni in STL_ERROR sono utili per la risoluzione di certi errori. Un tecnico dell' AWS assistenza potrebbe chiederti di fornire queste informazioni come parte del processo di risoluzione dei problemi.

STL_ERROR è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Per una lista di codici di errore che possono essere generati durante il caricamento dei dati con il comando Copy, consultare Riferimento per gli errori di caricamento.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
elaborazione	character(12)	Processo che ha generato l'eccezione.
recordtime	timestamp	Ora in cui l'errore si è verificato.

Nome colonna	Tipo di dati	Descrizione
pid	integer	ID processo. La tabella STL_QUERY contiene gli ID di processo e gli ID di query univoci per le query completate.
errcode	integer	Codice di errore che corrisponde alla categoria di errore.
file	character(90)	Nome del file di origine in cui si è verificato l'errore.
linenum	integer	Numero di riga nel file di origine in cui si è verificato l'errore.
context	character(100)	Causa dell'errore.
error	character(512)	Messaggio di errore.

Query di esempio

L'esempio seguente illustra come recuperare l'informazione di errore da STL_ERROR.

```
select process, errcode, linenum as line,
trim(error) as err
from stl_error;

   process    | errcode | line |                                err
--------------+---------+------
+-----------------------------------------------------------------
 padbmaster   |    8001 |  194 | Path prefix: s3://redshift-downloads/testnulls/
venue.txt*
 padbmaster   |    8001 |  529 | Listing bucket=redshift-downloads prefix=tests/
category-csv-quotes
 padbmaster   |       2 |  190 | database "template0" is not currently accepting
 connections
 padbmaster   |      32 | 1956 | pq_flush: could not send data to client: Broken pipe
(4 rows)
```

STL_EXPLAIN

Mostra il piano EXPLAIN per una query inviata in esecuzione.

STL_EXPLAIN è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_EXPLAIN contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
nodeid	integer	Identificatore di nodo di piano, dove un nodo corrisponde a una o più fasi nell'esecuzione della query.
parentid	integer	Identificatore di nodo di piano per un nodo padre. Un nodo padre ha un certo numero di nodi figli. Ad esempio, un merge join è il padre delle scansioni sulle tabelle collegate.
plannode	character(400)	Il testo del nodo dall'output EXPLAIN. I nodi di piano riferiti all'esecuzione sui nodi di calcolo hanno il prefisso **XN** nell'output EXPLAIN.
info	character(400)	Informazioni di qualificatore e di filtro per il nodo di piano. Ad esempio, le condizioni di join e le restrizioni di clausola WHERE sono incluse in questa colonna.

Query di esempio

Considera il seguente output EXPLAIN per una query di join d'aggregazione:

```
explain select avg(datediff(day, listtime, saletime)) as avgwait
from sales, listing where sales.listid = listing.listid;
                                QUERY PLAN

------------------------------------------------------------------------------
 XN Aggregate  (cost=6350.30..6350.31 rows=1 width=16)
  ->  XN Hash Join DS_DIST_NONE  (cost=47.08..6340.89 rows=3766 width=16)
        Hash Cond: ("outer".listid = "inner".listid)
        -> XN Seq Scan on listing  (cost=0.00..1924.97 rows=192497 width=12)
        -> XN Hash  (cost=37.66..37.66 rows=3766 width=12)
              -> XN Seq Scan on sales  (cost=0.00..37.66 rows=3766 width=12)
(6 rows)
```

Se esegui questa query e l'ID di query è 10, è possibile utilizzare la tabella STL_EXPLAIN per visualizzare lo stesso tipo di informazioni restituito dal comando EXPLAIN:

```
select query,nodeid,parentid,substring(plannode from 1 for 30),
substring(info from 1 for 20) from stl_explain
where query=10 order by 1,2;

query| nodeid |parentid|           substring            |    substring
-----+--------+--------+--------------------------------+-------------------
10   |      1 |      0 |XN Aggregate  (cost=6717.61..6  |
10   |      2 |      1 |  -> XN Merge Join DS_DIST_NO   | Merge Cond:("outer"
10   |      3 |      2 |       -> XN Seq Scan on lis    |
10   |      4 |      2 |       -> XN Seq Scan on sal    |
(4 rows)
```

Considera la query seguente:

```
select event.eventid, sum(pricepaid)
from event, sales
where event.eventid=sales.eventid
group by event.eventid order by 2 desc;

eventid |   sum
--------+----------
    289 | 51846.00
```

```
   7895 | 51049.00
   1602 | 50301.00
    851 | 49956.00
   7315 | 49823.00
...
```

Se l'ID di questa query è 15, la seguente query sulla vista di sistema restituisce i nodi del piano che sono stati completati. In questo caso, l'ordine dei nodi è invertito per mostrare l'effettivo ordine di esecuzione:

```
select query,nodeid,parentid,substring(plannode from 1 for 56)
from stl_explain where query=15 order by 1, 2 desc;

query|nodeid|parentid|                          substring
-----+------+--------+-----------------------------------------------------------
15   |    8 |      7 |                               -> XN Seq Scan on eve
15   |    7 |      5 |                          -> XN Hash(cost=87.98..87.9
15   |    6 |      5 |                          -> XN Seq Scan on sales(cos
15   |    5 |      4 |                    -> XN Hash Join DS_DIST_OUTER(cos
15   |    4 |      3 |              -> XN HashAggregate(cost=862286577.07..
15   |    3 |      2 |        -> XN Sort(cost=1000862287175.47..10008622871
15   |    2 |      1 |  -> XN Network(cost=1000862287175.47..1000862287197.
15   |    1 |      0 |XN Merge(cost=1000862287175.47..1000862287197.46 rows=87
(8 rows)
```

La seguente query recupera gli ID di query per tutti i piani di query che contengono una funzione finestra:

```
select query, trim(plannode) from stl_explain
where plannode like '%Window%';

query|                                  btrim
-----+-------------------------------------------------------------------------
26   | -> XN Window(cost=1000985348268.57..1000985351256.98 rows=170 width=33)
27   | -> XN Window(cost=1000985348268.57..1000985351256.98 rows=170 width=33)
(2 rows)
```

STL_FILE_SCAN

Restituisce i file che Amazon Redshift legge durante il caricamento dei dati per mezzo del comando COPY.

L'esecuzione di query su questa tabella può essere utile nella risoluzione degli errori di caricamento dei dati. STL_FILE_SCAN può essere molto utile nell'identificazione dei problemi nei caricamenti di dati in parallelo perché in genere questo tipo di caricamenti caricano molti file con un unico comando COPY.

STL_FILE_SCAN è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_FILE_SCAN contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_LOAD_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
nome	character(90)	Percorso completo e nome del file caricato.
lines	bigint	Numero di righe lette dal file.
byte	bigint	Numero di byte letti dal file.
loadtime	bigint	Periodo di tempo impiegato a caricare il file (in microsecondi).

Nome colonna	Tipo di dati	Descrizione
curtime	Timestamp	Timestamp corrispondente all'ora in cui Amazon Redshift ha iniziato a elaborare il file.
is_partial	integer	Valore che se true (1) indica che il file di input viene diviso in intervalli durante un'operazione COPY. Se il valore è false (0), il file di input non viene diviso.
start_offset	bigint	Valore che, se il file di input viene diviso durante un'operaz ione COPY, indica il valore di offset della divisione (in byte). Se il file non è diviso, questo valore è 0.

Query di esempio

La seguente query recupera i nomi e i tempi di caricamento di tutti i file che hanno impiegato più di 1.000.000 microsecondi a essere letti da Amazon Redshift.

```
select trim(name)as name, loadtime from stl_file_scan
where loadtime > 1000000;
```

Questa query restituisce il seguente output di esempio.

```
           name             | loadtime
----------------------------+----------
 listings_pipe.txt          |  9458354
 allusers_pipe.txt          |  2963761
 allevents_pipe.txt         |  1409135
 tickit/listings_pipe.txt   |  7071087
 tickit/allevents_pipe.txt  |  1237364
 tickit/allusers_pipe.txt   |  2535138
 listings_pipe.txt          |  6706370
 allusers_pipe.txt          |  3579461
 allevents_pipe.txt         |  1313195
 tickit/allusers_pipe.txt   |  3236060
 tickit/listings_pipe.txt   |  4980108
(11 rows)
```

STL_HASH

Analizza le fasi di esecuzione di hash per le query.

STL_HASH è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_HASH contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .

Nome colonna	Tipo di dati	Descrizione
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
byte	bigint	Dimensione, in byte, di tutte le righe di output della fase.
slots	integer	Numero totale di bucket di hash.
occupied	integer	Numero totale di slot che contengono record.
maxlength	integer	Dimensioni dello slot più grande.
tbl	integer	ID tabella.
is_diskbased	character(1)	Se true (t), la query è stata eseguita come un'operazione basata su disco. Se false (f), la query è stata eseguita in memoria.
workmem	bigint	Numero totale di byte della memoria di lavoro assegnati alla fase.
num_parts	integer	Numero totale di partizioni in cui è stata divisa una tabella di hash durante una fase di hash.
est_rows	bigint	Numero di righe previsto da sottoporre a hash.
num_blocks_permitted	integer	Queste informazioni sono solo per uso interno.
resizes	integer	Queste informazioni sono solo per uso interno.

Nome colonna	Tipo di dati	Descrizione
checksum	bigint	Queste informazioni sono solo per uso interno.
runtime_filter_size	integer	Dimensioni del filtro di runtime in byte.
max_runtime_filter_size	integer	Dimensioni massime del filtro di runtime in byte.

Query di esempio

L'esempio seguente restituisce informazioni sul numero di partizioni utilizzate in un hash per la query 720 e indica che nessuna delle fasi è stata eseguita sul disco.

```
select slice, rows, bytes, occupied, workmem, num_parts, est_rows,
 num_blocks_permitted, is_diskbased
from stl_hash
where query=720 and segment=5
order by slice;
```

```
 slice | rows | bytes  | occupied | workmem  | num_parts | est_rows |
 num_blocks_permitted | is_diskbased
-------+------+--------+----------+----------+-----------+----------
+----------------------+--------------
     0 |  145 | 585800 |        1 | 88866816 |        16 |        1 |
 52                   f
     1 |    0 |      0 |        0 |        0 |        16 |        1 |
 52                   f
(2 rows)
```

STL_HASHJOIN

Analizza le fasi di esecuzione di hash join per le query.

STL_HASHJOIN è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_HASHJOIN contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.

Nome colonna	Tipo di dati	Descrizione
righe	bigint	Numero totale di righe elaborate.
tbl	integer	ID tabella.
num_parts	integer	Numero totale di partizioni in cui è stata divisa una tabella di hash durante una fase di hash.
join_type	integer	Il tipo di join per la fase: • 0. La query ha utilizzato un inner join. • 1. La query ha utilizzato un left outer join. • 2. La query ha utilizzato un full outer join. • 3. La query ha utilizzato un right outer join. • 4. La query ha utilizzato un operatore UNION. • 5. La query ha utilizzato una condizione IN. • 6. Queste informazioni sono solo per uso interno. • 7. Queste informazioni sono solo per uso interno. • 8. Queste informazioni sono solo per uso interno. • 9. Queste informazioni sono solo per uso interno. • 10. Queste informazioni sono solo per uso interno. • 11. Queste informazioni sono solo per uso interno. • 12. Queste informazioni sono solo per uso interno.
hash_looped	character(1)	Queste informazioni sono solo per uso interno.
switched_parts	character(1)	Indica se i lati compilazione (o esterno) e probe (o interno) sono stati scambiati.
used_prefetching	character(1)	Queste informazioni sono solo per uso interno.

Nome colonna	Tipo di dati	Descrizione
hash_segment	integer	Il segmento della fase di hash corrispondente.
hash_step	integer	Il numero di fase della fase di hash corrispondente.
checksum	bigint	Queste informazioni sono solo per uso interno.
distribuzione	integer	Queste informazioni sono solo per uso interno.

Query di esempio

L'esempio seguente restituisce il numero di partizioni utilizzate in un hash join per la query 720.

```
select query, slice, tbl, num_parts
from stl_hashjoin
where query=720 limit 10;
```

```
 query | slice | tbl | num_parts
-------+-------+-----+-----------
   720 |     0 | 243 |         1
   720 |     1 | 243 |         1
(2 rows)
```

STL_INSERT

Analizza le fasi di esecuzione di inserimento per le query.

STL_INSERT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_INSERT contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite

sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
tbl	integer	ID tabella.

Nome colonna	Tipo di dati	Descrizione
inserted_mega_value	character(1)	Queste informazioni sono solo per uso interno. Queste informazioni mostrano se la fase di inserimento specifica ta ha inserito un valore elevato. Un valore elevato verrà archiviato in più blocchi. La dimensione del blocco è 1 MB per impostazione predefinita, un valore elevato è superiore a 1 MB in un'impostazione predefinita.

Query di esempio

L'esempio seguente restituisce le fasi di esecuzione di inserimento per la query più recente.

```
select slice, segment, step, tasknum, rows, tbl
from stl_insert
where query=pg_last_query_id();
```

```
 slice | segment | step | tasknum | rows  |  tbl
-------+---------+------+---------+-------+--------
     0 |       2 |    2 |      15 | 24958 | 100548
     1 |       2 |    2 |      15 | 25032 | 100548
(2 rows)
```

STL_LIMIT

Analizza le fasi di esecuzione che si verificano quando una clausola LIMIT è utilizzata in una query SELECT.

STL_LIMIT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_LIMIT contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista

di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
checksum	bigint	Queste informazioni sono solo per uso interno.

Query di esempio

Al fine di generare una riga in STL_LIMIT, questo esempio esegue prima la query seguente sulla tabella VENUE utilizzando la clausola LIMIT.

```
select * from venue
order by 1
limit 10;
```

```
 venueid |         venuename          |    venuecity    | venuestate | venueseats
---------+----------------------------+-----------------+------------+------------
       1 | Toyota Park                | Bridgeview      | IL         |          0
       2 | Columbus Crew Stadium      | Columbus        | OH         |          0
       3 | RFK Stadium                | Washington      | DC         |          0
       4 | CommunityAmerica Ballpark  | Kansas City     | KS         |          0
       5 | Gillette Stadium           | Foxborough      | MA         |      68756
       6 | New York Giants Stadium    | East Rutherford | NJ         |      80242
       7 | BMO Field                  | Toronto         | ON         |          0
       8 | The Home Depot Center      | Carson          | CA         |          0
       9 | Dick's Sporting Goods Park | Commerce City   | CO         |          0
      10 | Pizza Hut Park             | Frisco          | TX         |          0
(10 rows)
```

Successivamente, esegui la seguente query per trovare l'ID di query dell'ultima query che hai eseguito sulla tabella VENUE.

```
select max(query)
from stl_query;
```

```
  max
--------
 127128
(1 row)
```

A scelta, è possibile eseguire la seguente query per verificare che l'ID di query corrisponda alla query LIMIT che hai eseguito in precedenza.

```
select query, trim(querytxt)
from stl_query
where query=127128;
```

```
 query  |                  btrim
--------+-----------------------------------------
 127128 | select * from venue order by 1 limit 10;
(1 row)
```

Infine, esegui la seguente query per restituire le informazioni sulla query LIMIT della tabella STL_LIMIT.

```
select slice, segment, step, starttime, endtime, tasknum
from stl_limit
where query=127128
order by starttime, endtime;
```

```
  slice | segment | step |         starttime          |          endtime           |
 tasknum
 -------+---------+------+----------------------------+----------------------------
+---------
      1 |       1 |    3 | 2013-09-06 22:56:43.608114 | 2013-09-06 22:56:43.609383 |
   15
      0 |       1 |    3 | 2013-09-06 22:56:43.608708 | 2013-09-06 22:56:43.609521 |
   15
  10000 |       2 |    2 | 2013-09-06 22:56:43.612506 | 2013-09-06 22:56:43.612668 |
    0
(3 rows)
```

STL_LOAD_COMMITS

Restituisce informazioni per tracciare o risolvere i problemi di caricamento dati.

Questa vista registra il progresso di tutti i file di dati mentre vengono caricati in una tabella del database.

STL_LOAD_COMMITS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_LOAD_COMMITS contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di

monitoraggio SYS SYS_LOAD_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Sezione caricata per questa voce.
nome	character(256)	Valore definito dal sistema.
filename	character(256)	Nome del file tracciato.
byte_offset	integer	Queste informazioni sono solo per uso interno.
lines_scanned	integer	Numero di righe analizzate dal file di importazione. Questo numero può non corrispondere al numero di righe effettivamente caricate. Ad esempio, il carico può analizzare e tollerare un numero di record danneggiati, basato sull'opzione MAXERROR nel comando COPY.
errors	integer	Queste informazioni sono solo per uso interno.
curtime	timestamp	L'ultima volta in cui la voce è stata aggiornata.
status	integer	Queste informazioni sono solo per uso interno.
file_format	character(16)	Formato del file di importazione. I valori possibili sono i seguenti: • Avro • JSON • ORC

Nome colonna	Tipo di dati	Descrizione
		• Parquet • Testo
is_partial	integer	Valore che se true (1) indica che il file di input viene diviso in intervalli durante un'operazione COPY. Se il valore è false (0), il file di input non viene diviso.
start_offset	bigint	Valore che, se il file di input viene diviso durante un'operaz ione COPY, indica il valore di offset della divisione (in byte). Ogni divisione di file viene registrata come record separato con il corrispondente valore start_offset. Se il file non è diviso, questo valore è 0.
copy_job_id	bigint	Identificatore del processo di copia. Il valore `0` indica l'assenza del processo.

Query di esempio

L'esempio seguente restituisce i dettagli per l'ultima operazione COPY.

```
select query, trim(filename) as file, curtime as updated
from stl_load_commits
where query = pg_last_copy_id();

 query |               file                |          updated
-------+-----------------------------------+----------------------------
 28554 | s3://dw-tickit/category_pipe.txt | 2013-11-01 17:14:52.648486
(1 row)
```

La query seguente contiene voci per un nuovo carico delle tabelle nel database TICKIT:

```
select query, trim(filename), curtime
from stl_load_commits
where filename like '%tickit%' order by query;
```

```
 query |           btrim           |          curtime
```

```
-------+--------------------------+--------------------------
 22475 | tickit/allusers_pipe.txt  | 2013-02-08 20:58:23.274186
 22478 | tickit/venue_pipe.txt     | 2013-02-08 20:58:25.070604
 22480 | tickit/category_pipe.txt  | 2013-02-08 20:58:27.333472
 22482 | tickit/date2008_pipe.txt  | 2013-02-08 20:58:28.608305
 22485 | tickit/allevents_pipe.txt | 2013-02-08 20:58:29.99489
 22487 | tickit/listings_pipe.txt  | 2013-02-08 20:58:37.632939
 22593 | tickit/allusers_pipe.txt  | 2013-02-08 21:04:08.400491
 22596 | tickit/venue_pipe.txt     | 2013-02-08 21:04:10.056055
 22598 | tickit/category_pipe.txt  | 2013-02-08 21:04:11.465049
 22600 | tickit/date2008_pipe.txt  | 2013-02-08 21:04:12.461502
 22603 | tickit/allevents_pipe.txt | 2013-02-08 21:04:14.785124
 22605 | tickit/listings_pipe.txt  | 2013-02-08 21:04:20.170594

(12 rows)
```

Il fatto che un record sia scritto nel file di log per questa vista di sistema non significa che il carico abbia eseguito il commit in modo efficiente come parte della transazione contenente. Per verificare i commit di carico, esegui una query sulla vista STL_UTILITYTEXT e cerca il record COMMIT che corrisponde a una transazione COPY. Ad esempio, questa query collega STL_LOAD_COMMITS e STL_QUERY sulla base di una subquery sulla STL_UTILITYTEXT:

```
select l.query,rtrim(l.filename),q.xid
from stl_load_commits l, stl_query q
where l.query=q.query
and exists
(select xid from stl_utilitytext where xid=q.xid and rtrim("text")='COMMIT');

 query |           rtrim           |  xid
-------+---------------------------+-------
 22600 | tickit/date2008_pipe.txt  | 68311
 22480 | tickit/category_pipe.txt  | 68066
  7508 | allusers_pipe.txt         | 23365
  7552 | category_pipe.txt         | 23415
  7576 | allevents_pipe.txt        | 23429
  7516 | venue_pipe.txt            | 23390
  7604 | listings_pipe.txt         | 23445
 22596 | tickit/venue_pipe.txt     | 68309
 22605 | tickit/listings_pipe.txt  | 68316
 22593 | tickit/allusers_pipe.txt  | 68305
 22485 | tickit/allevents_pipe.txt | 68071
  7561 | allevents_pipe.txt        | 23429
  7541 | category_pipe.txt         | 23415
```

```
  7558 | date2008_pipe.txt         | 23428
 22478 | tickit/venue_pipe.txt     | 68065
   526 | date2008_pipe.txt         |  2572
  7466 | allusers_pipe.txt         | 23365
 22482 | tickit/date2008_pipe.txt  | 68067
 22598 | tickit/category_pipe.txt  | 68310
 22603 | tickit/allevents_pipe.txt | 68315
 22475 | tickit/allusers_pipe.txt  | 68061
   547 | date2008_pipe.txt         |  2572
 22487 | tickit/listings_pipe.txt  | 68072
  7531 | venue_pipe.txt            | 23390
  7583 | listings_pipe.txt         | 23445
(25 rows)
```

Gli esempi seguenti evidenziano i valori delle colonne is_partial e start_offset.

```
-- Single large file copy without scan range
SELECT count(*) FROM stl_load_commits WHERE query = pg_last_copy_id();
1

-- Single large uncompressed, delimited file copy with scan range
SELECT count(*) FROM stl_load_commits WHERE query = pg_last_copy_id();
16

-- Scan range offset logging in the file at 64MB boundary.
SELECT start_offset FROM stl_load_commits
WHERE query = pg_last_copy_id() ORDER BY start_offset;
0
67108864
134217728
201326592
268435456
335544320
402653184
469762048
536870912
603979776
671088640
738197504
805306368
872415232
939524096
1006632960
```

STL_LOAD_ERRORS

Visualizza i record di tutti gli errori di caricamento di Amazon Redshift.

STL_LOAD_ERRORS contiene una cronologia di tutti gli errori di caricamento di Amazon Redshift. Consultare Riferimento per gli errori di caricamento per un elenco completo dei possibili errori di caricamento e le spiegazioni.

Dopo aver interrogato STL_LOAD_ERRORS per trovare informazioni generiche sull'errore, esegui una query in STL_LOADERROR_DETAIL per dettagli aggiuntivi, come la riga e la colonna di dati esatte in cui si è verificato l'errore di analisi.

STL_LOAD_ERRORS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_LOAD_ERRORS contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_LOAD_ERROR_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
sezione	integer	Sezione in cui si è verificato l'errore.
tbl	integer	ID tabella.
starttime	timestamp	Ora di inizio in UTC per il carico.
session	integer	ID di sessione per la sessione che esegue il carico.

Nome colonna	Tipo di dati	Descrizione
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
filename	character(256)	Percorso completo verso il file di input per il carico.
line_number	bigint	Numero di riga nel file di importazione con l'errore. Per COPY da JSON, il numero di riga dell'ultima riga dell'oggetto JSON con l'errore.
colname	character(127)	Campo con l'errore.
tipo	character(10)	Tipo di dati del campo.
col_length	character(10)	Lunghezza della colonna, se applicabile. Questo campo è popolato quando il tipo di dati ha una lunghezza limite. Ad esempio, per una colonna con un tipo di dati di "character(3)", questa colonna conterrà il valore "3".
posizione	integer	Posizione dell'errore nel campo.
raw_line	character(1024)	Dati di caricamento non elaborati che contengono l'errore. I caratteri multibyte nei dati di caricamento sono sostituiti da un punto.
raw_field_value	char(1024)	Valore di preanalisi per il campo "colname" che ha portato all'errore di analisi.
err_code	integer	Codice di errore.
err_reason	character(100)	Spiegazione dell'errore.
is_partial	integer	Valore che se true (1) indica che il file di input viene diviso in intervalli durante un'operazione COPY. Se il valore è false (0), il file di input non viene diviso.

Nome colonna	Tipo di dati	Descrizione
start_offset	bigint	Valore che, se il file di input viene diviso durante un'operazione COPY, indica il valore di offset della divisione (in byte). Se il numero di riga nel file è sconosciuto, il numero di riga è -1. Se il file non è diviso, questo valore è 0.
copy_job_id	bigint	Identificatore del processo di copia. Il valore `0` indica l'assenza del processo.

Query di esempio

La query seguente collega STL_LOAD_ERRORS a STL_LOADERROR_DETAIL per visualizzare gli errori dei dettagli che si sono verificati durante il caricamento più recente.

```
select d.query, substring(d.filename,14,20),
d.line_number as line,
substring(d.value,1,16) as value,
substring(le.err_reason,1,48) as err_reason
from stl_loaderror_detail d, stl_load_errors le
where d.query = le.query
and d.query = pg_last_copy_id();

 query |    substring      | line |  value   |             err_reason
-------+-------------------+------+----------+----------------------------
    558| allusers_pipe.txt |  251 | 251      | String contains invalid or
                                              unsupported UTF8 code
    558| allusers_pipe.txt |  251 | ZRU29FGR | String contains invalid or
                                              unsupported UTF8 code
    558| allusers_pipe.txt |  251 | Kaitlin  | String contains invalid or
                                              unsupported UTF8 code
    558| allusers_pipe.txt |  251 | Walter   | String contains invalid or
                                              unsupported UTF8 code
```

L'esempio seguente utilizza STL_LOAD_ERRORS con STV_TBL_PERM per creare una nuova visualizzazione e la utilizza per determinare quale errore si è verificato durante il caricamento dei dati nella tabella EVENT:

```
create view loadview as
```

```
(select distinct tbl, trim(name) as table_name, query, starttime,
trim(filename) as input, line_number, colname, err_code,
trim(err_reason) as reason
from stl_load_errors sl, stv_tbl_perm sp
where sl.tbl = sp.id);
```

Successivamente, la seguente query restituisce l'ultimo errore che si è verificato durante il caricamento della tabella EVENT:

```
select table_name, query, line_number, colname, starttime,
trim(reason) as error
from loadview
where table_name ='event'
order by line_number limit 1;
```

La query restituisce l'ultimo errore di caricamento che si è verificato nella tabella EVENT. Se non si sono verificati errori di caricamento, la query restituisce zero righe. In questo esempio, la query restituisce un solo errore:

```
 table_name | query | line_number | colname | error | starttime
------+-----+----+----+-------------------------------------------------------
+----------------------
event | 309 |  0 |  5 | Error in Timestamp value or format [%Y-%m-%d %H:%M:%S] |
 2014-04-22 15:12:44

(1 row)
```

Nei casi in cui il comando COPY divide automaticamente dati di file di grandi dimensioni, non compressi e delimitati da testo per facilitare il parallelismo, le colonne line_number,is_partial, e start_offset le colonne mostrano informazioni relative alle divisioni. (Il numero di riga può essere sconosciuto nei casi in cui il numero di riga del file originale non è disponibile.)

```
--scan ranges information
SELECT line_number, POSITION, btrim(raw_line), btrim(raw_field_value),
btrim(err_reason), is_partial, start_offset FROM stl_load_errors
WHERE query = pg_last_copy_id();

--result
-1,51,"1008771|13463413|463414|2|28.00|38520.72|0.06|0.07|NO|1998-08-30|1998-09-25|
1998-09-04|TAKE BACK RETURN|RAIL|ans cajole sly","NO","Char length exceeds DDL
 length",1,67108864
```

STL_LOADERROR_DETAIL

Mostra un log degli errori di analisi dei dati che si sono verificati durante l'utilizzo di un comando COPY nelle tabelle di caricamento. Per preservare spazio su disco, per ogni operazione di caricamento vengono registrati un massimo di 20 errori per ogni sezione di nodo.

Quando Amazon Redshift non riesce ad analizzare un campo in una riga di dati mentre viene caricato in una tabella, si verifica un errore di analisi. Ad esempio, se una colonna della tabella prevede un tipo di dati di numeri interi e in quel campo il file di dati contiene una stringa di lettere, ciò provoca un errore di analisi.

Dopo aver eseguito una query su STL_LOAD_ERRORS per trovare informazioni generali sull'errore, esegui una query su STL_LOADERROR_DETAIL per dettagli aggiuntivi, ad esempio la riga e la colonna di dati esatte in cui si è verificato un errore di analisi.

La vista STL_LOADERROR_DETAIL contiene tutte le colonne di dati, inclusa la colonna in cui si è verificato l'errore di analisi e quelle precedenti. Utilizza il campo VALUE per visualizzare il valore di dati effettivamente analizzato in questa colonna, incluse le colonne che sono state analizzate correttamente prima dell'errore.

Questa vista è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_LOADERROR_DETAIL contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_LOAD_ERROR_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.

Nome colonna	Tipo di dati	Descrizione
sezione	integer	Sezione in cui si è verificato l'errore.
session	integer	ID di sessione per la sessione che esegue il carico.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
filename	character(256)	Percorso completo verso il file di input per il carico.
line_number	bigint	Numero di riga nel file di importazione con l'errore.
field	integer	Campo con l'errore.
colname	character(1024)	Nome della colonna.
value	character(1024)	Valore dei dati del campo analizzati. (Può essere troncato). I caratteri multibyte nei dati di caricamento sono sostituiti da un punto.
is_null	integer	Se il valore analizzato è null o meno.
tipo	character(10)	Tipo di dati del campo.
col_length	character(10)	Lunghezza della colonna, se applicabile. Questo campo è popolato quando il tipo di dati ha una lunghezza limite. Ad esempio, per una colonna con un tipo di dati di "character(3)", questa colonna conterrà il valore "3".

Query di esempio

La query seguente collega STL_LOAD_ERRORS a STL_LOADERROR_DETAIL per visualizzare i dettagli di un errore di analisi verificatosi durante il caricamento della tabella EVENT, che ha un ID di tabella di 100133:

```
select d.query, d.line_number, d.value,
le.raw_line, le.err_reason
```

```
from stl_loaderror_detail d, stl_load_errors le
where
d.query = le.query
and tbl = 100133;
```

Il seguente output di esempio mostra le colonne il cui caricamento è riuscito, inclusa la colonna con l'errore. In questo esempio, in due colonne il caricamento è riuscito prima che si verificasse l'errore di analisi nella terza colonna, in cui una stringa di caratteri è stata erroneamente analizzata per un campo che prevedeva un integer. Poiché il campo prevedeva un integer, ha analizzato la stringa "aaa", che sono dati non inizializzati, come null e ha generato un errore di analisi. L'output mostra il valore non elaborato, il valore analizzato e il motivo dell'errore:

```
query  | line_number | value | raw_line | err_reason
-------+-------------+-------+----------+----------------
4      |      3      |  1201 |  1201    | Invalid digit
4      |      3      |   126 |   126    | Invalid digit
4      |      3      |       |   aaa    | Invalid digit
(3 rows)
```

Quando una query collega STL_LOAD_ERRORS e STL_LOADERROR_DETAIL, mostra un motivo di errore per ogni colonna nella riga di dati, il che significa semplicemente che si è verificato un errore in quella riga. L'ultima riga nei risultati è la colonna in cui si è effettivamente verificato l'errore.

STL_MERGE

Analizza le fasi di esecuzione di unione per le query. Queste fasi si verificano quando i risultati di operazioni parallele (come gli ordinamenti e i join) sono unite per una successiva elaborazione.

STL_MERGE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_MERGE contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.

Query di esempio

L'esempio seguente restituisce 10 risultati di esecuzione di unione.

```
select query, step, starttime, endtime, tasknum, rows
from stl_merge
limit 10;
```

```
 query | step |       starttime     |        endtime      | tasknum | rows
-------+------+---------------------+---------------------+---------+------
     9 |    0 | 2013-08-12 20:08:14 | 2013-08-12 20:08:14 |       0 |    0
    12 |    0 | 2013-08-12 20:09:10 | 2013-08-12 20:09:10 |       0 |    0
    15 |    0 | 2013-08-12 20:10:24 | 2013-08-12 20:10:24 |       0 |    0
    20 |    0 | 2013-08-12 20:11:27 | 2013-08-12 20:11:27 |       0 |    0
    26 |    0 | 2013-08-12 20:12:28 | 2013-08-12 20:12:28 |       0 |    0
    32 |    0 | 2013-08-12 20:14:33 | 2013-08-12 20:14:33 |       0 |    0
    38 |    0 | 2013-08-12 20:16:43 | 2013-08-12 20:16:43 |       0 |    0
    44 |    0 | 2013-08-12 20:17:05 | 2013-08-12 20:17:05 |       0 |    0
    50 |    0 | 2013-08-12 20:18:48 | 2013-08-12 20:18:48 |       0 |    0
    56 |    0 | 2013-08-12 20:20:48 | 2013-08-12 20:20:48 |       0 |    0
(10 rows)
```

STL_MERGEJOIN

Analizza le fasi di esecuzione di merge join per le query.

STL_MERGEJOIN è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_MERGEJOIN contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.

Nome colonna	Tipo di dati	Descrizione
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
tbl	integer	ID tabella. Questo è l'ID per la tabella interna che è stata utilizzata nel merge join.
checksum	bigint	Queste informazioni sono solo per uso interno.

Query di esempio

L'esempio seguente restituisce i risultati di merge join per la query più recente.

```
select sum(s.qtysold), e.eventname
from event e, listing l, sales s
where e.eventid=l.eventid
and l.listid= s.listid
group by e.eventname;
```

```
select * from stl_mergejoin where query=pg_last_query_id();
```

```
 userid | query | slice | segment | step |         starttime   |          endtime    |
 tasknum | rows | tbl
--------+-------+-------+---------+------+---------------------+---------------------
+---------+------+-----
    100 | 27399 |     3 |       4 |    4 | 2013-10-02 16:30:41 | 2013-10-02 16:30:41 |
     19 |43428 | 240
    100 | 27399 |     0 |       4 |    4 | 2013-10-02 16:30:41 | 2013-10-02 16:30:41 |
     19 |43159 | 240
    100 | 27399 |     2 |       4 |    4 | 2013-10-02 16:30:41 | 2013-10-02 16:30:41 |
     19 |42778 | 240
    100 | 27399 |     1 |       4 |    4 | 2013-10-02 16:30:41 | 2013-10-02 16:30:41 |
     19 |43091 | 240
```

STL_MV_STATE

La tabella vista STL_MV_STATE contiene una riga per ogni transizione di stato di una vista materializzata.

Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

STL_MV_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_MV_STATE. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	bigint	L'ID dell'utente che ha creato l'evento.
starttime	timestamp	L'orario di inizio dell'evento.
xid	bigint	L'ID di transazione dell'evento.

Nome colonna	Tipo di dati	Descrizione
event_desc	char(500)	L'evento che ha richiesto la modifica dello stato. Esempi di valori possibili includono quanto segue: • Il tipo di colonna è stato modificato • La colonna è stata eliminata • La colonna è stata rinominata • Il nome dello schema è stato modificato • Piccola conversione tabella • TRUNCATE • Vacuum Nota che ci sono altri valori possibili per questa colonna.
db_name	char(128)	Il database che contiene la vista materializzata.
base_table_schema	char(128)	Lo schema della tabella di base.
base_table_name	char(128)	Il nome della tabella di base.
mv_schema	char(128)	Lo schema della vista materializzata.
mv_name	char(128)	Il nome della vista materializzata.
stato	character(32)	Lo stato modificato della vista materializzata come descritto di seguito: • Ricalcola • Non aggiornabile

La tabella seguente mostra combinazioni di esempio di `event_desc` e `state`.

```
          event_desc     |     state
-------------------------+---------------
 TRUNCATE                | Recompute
 TRUNCATE                | Recompute
 Small table conversion  | Recompute
 Vacuum                  | Recompute
 Column was renamed      | Unrefreshable
 Column was dropped      | Unrefreshable
 Table was renamed       | Unrefreshable
 Column type was changed | Unrefreshable
 Schema name was changed | Unrefreshable
```

Query di esempio

Per visualizzare il log delle transizioni di stato delle viste materializzate, eseguire la query seguente.

```
select * from stl_mv_state;
```

Questa query restituisce il seguente output di esempio:

```
 userid |         starttime          | xid  |            event_desc       | db_name |
 base_table_schema   |   base_table_name    |      mv_schema       | mv_name       |
  state
--------+----------------------------+------+-----------------------------+---------
+---------------------+---------------------+---------------------+--------------
+---------------
    138 | 2020-02-14 02:21:25.578885 | 5180 | TRUNCATE                    | dev     |
 public              | mv_base_table        | public              | mv_test       |
 Recompute
    138 | 2020-02-14 02:21:56.846774 | 5275 | Column was dropped          | dev     |
                     | mv_base_table        | public              | mv_test       |
 Unrefreshable
    100 | 2020-02-13 22:09:53.041228 | 1794 | Column was renamed          | dev     |
                     | mv_base_table        | public              | mv_test       |
 Unrefreshable
      1 | 2020-02-13 22:10:23.630914 | 1893 | ALTER TABLE ALTER SORTKEY   | dev     |
 public              | mv_base_table_sorted | public              | mv_test       |
 Recompute
      1 | 2020-02-17 22:57:22.497989 | 8455 | ALTER TABLE ALTER DISTSTYLE | dev     |
 public              | mv_base_table        | public              | mv_test       |
 Recompute
```

```
   173 | 2020-02-17 22:57:23.591434 | 8504 | Table was renamed           | dev     |
                    | mv_base_table        | public               | mv_test       |
Unrefreshable
   173 | 2020-02-17 22:57:27.229423 | 8592 | Column type was changed     | dev     |
                    | mv_base_table        | public               | mv_test       |
Unrefreshable
   197 | 2020-02-17 22:59:06.212569 | 9668 | TRUNCATE                    | dev     |
schemaf796e415850f4f | mv_base_table        | schemaf796e415850f4f | mv_test       |
Recompute
   138 | 2020-02-14 02:21:55.705655 | 5226 | Column was renamed          | dev     |
                    | mv_base_table        | public               | mv_test       |
Unrefreshable
     1 | 2020-02-14 02:22:26.292434 | 5325 | ALTER TABLE ALTER SORTKEY   | dev     |
public              | mv_base_table_sorted | public               | mv_test       |
Recompute
```

STL_NESTLOOP

Analizza le fasi di esecuzione di loop nidificato per le query.

STL_NESTLOOP è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_NESTLOOP contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.

Nome colonna	Tipo di dati	Descrizione
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
tbl	integer	ID tabella.
checksum	bigint	Queste informazioni sono solo per uso interno.

Query di esempio

Poiché la query seguente non collega la tabella CATEGORY, genera un prodotto cartesiano parziale, che è sconsigliato. Qui viene mostrato un loop nidificato.

```
select count(event.eventname), event.eventname, category.catname, date.caldate
from event, category, date
where event.dateid = date.dateid
```

```
group by event.eventname, category.catname, date.caldate;
```

La seguente query mostra i risultati dalla query precedente nella vista STL_NESTLOOP.

```
select query, slice, segment as seg, step,
datediff(msec, starttime, endtime) as duration, tasknum, rows, tbl
from stl_nestloop
where query = pg_last_query_id();
```

```
 query | slice | seg | step | duration | tasknum | rows  | tbl
-------+-------+-----+------+----------+---------+-------+-----
  6028 |     0 |   4 |    5 |       41 |      22 | 24277 | 240
  6028 |     1 |   4 |    5 |       26 |      23 | 24189 | 240
  6028 |     3 |   4 |    5 |       25 |      23 | 24376 | 240
  6028 |     2 |   4 |    5 |       54 |      22 | 23936 | 240
```

STL_PARSE

Analizza le fasi di query che analizzano stringhe in valori binari per il caricamento.

STL_PARSE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_PARSE contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.

Nome colonna	Tipo di dati	Descrizione
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.

Query di esempio

L'esempio seguente restituisce tutti i risultati delle fasi di query per la sezione 1 e il segmento 0 in cui le stringhe sono state analizzate in valori binari.

```
select query, step, starttime, endtime, tasknum, rows
from stl_parse
where slice=1 and segment=0;
```

```
 query | step |     starttime       |        endtime      | tasknum |  rows
-------+------+---------------------+---------------------+---------+--------
```

```
   669 |    1 | 2013-08-12 22:35:13 | 2013-08-12 22:35:17 |      32 | 192497
   696 |    1 | 2013-08-12 22:35:49 | 2013-08-12 22:35:49 |      32 |      0
   525 |    1 | 2013-08-12 22:32:03 | 2013-08-12 22:32:03 |      13 |  49990
   585 |    1 | 2013-08-12 22:33:18 | 2013-08-12 22:33:19 |      13 |    202
   621 |    1 | 2013-08-12 22:34:03 | 2013-08-12 22:34:03 |      27 |    365
   651 |    1 | 2013-08-12 22:34:47 | 2013-08-12 22:34:53 |      35 | 192497
   590 |    1 | 2013-08-12 22:33:28 | 2013-08-12 22:33:28 |      19 |      0
   599 |    1 | 2013-08-12 22:33:39 | 2013-08-12 22:33:39 |      31 |     11
   675 |    1 | 2013-08-12 22:35:26 | 2013-08-12 22:35:27 |      38 |   3766
   567 |    1 | 2013-08-12 22:32:47 | 2013-08-12 22:32:48 |      23 |  49990
   630 |    1 | 2013-08-12 22:34:17 | 2013-08-12 22:34:17 |      36 |      0
   572 |    1 | 2013-08-12 22:33:04 | 2013-08-12 22:33:04 |      29 |      0
   645 |    1 | 2013-08-12 22:34:37 | 2013-08-12 22:34:38 |      29 |   8798
   604 |    1 | 2013-08-12 22:33:47 | 2013-08-12 22:33:47 |      37 |      0
(14 rows)
```

STL_PLAN_INFO

Utilizza la vista STL_PLAN_INFO per visualizzare l'output di EXPLAIN per una query in termini di un set di righe. Questo è un modo alternativo per visualizzare i piani di query.

STL_PLAN_INFO è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_PLAN_INFO contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.

Nome colonna	Tipo di dati	Descrizione
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
nodeid	integer	Identificatore di nodo di piano, dove un nodo corrisponde a una o più fasi nell'esecuzione della query.
segment	integer	Numero identificativo del segmento di query.
step	integer	Numero identificativo della fase di query.
locus	integer	Posizione in cui viene eseguita la fase. 0 se in un nodo di calcolo e 1 se nel nodo principale.
plannode	integer	Valore enumerato del nodo di piano. Consulta la tabella seguente per le enumerazioni del plannode. (La colonna PLANNODE in STL_EXPLAIN contiene il testo del nodo di piano.)
startupcost	double precision	Il costo relativo stimato della restituzione della prima riga di questa fase.
totalcost	double precision	Il costo relativo stimato dell'esecuzione della fase.
righe	bigint	Il numero stimato di righe che saranno prodotte dalla fase.
byte	bigint	Il numero stimato di byte che saranno prodotti dalla fase.

Query di esempio

Gli esempi seguenti confrontano i piani di query per una query SELECT semplice restituita utilizzando il comando EXPLAIN e interrogando la tabella STL_PLAN_INFO.

```
explain select * from category;
QUERY PLAN
------------------------------------------------------------
XN Seq Scan on category (cost=0.00..0.11 rows=11 width=49)
(1 row)
```

```
select * from category;
catid | catgroup | catname | catdesc
-------+----------+-----------+--------------------------------------------
1 | Sports | MLB | Major League Baseball
3 | Sports | NFL | National Football League
5 | Sports | MLS | Major League Soccer
...

select * from stl_plan_info where query=256;

query | nodeid | segment | step | locus | plannode | startupcost | totalcost
| rows | bytes
-------+--------+---------+------+-------+----------+-------------+-----------+------
+-------
256 | 1 | 0 | 1 | 0 | 104 | 0 | 0.11 | 11 | 539
256 | 1 | 0 | 0 | 0 | 104 | 0 | 0.11 | 11 | 539
(2 rows)
```

In questo esempio, PLANNODE 104 si riferisce alla scansione sequenziale della tabella CATEGORY.

```
select distinct eventname from event order by 1;

eventname
----------------------------------------------------------------------
.38 Special
3 Doors Down
70s Soul Jam
A Bronx Tale
...

explain select distinct eventname from event order by 1;

QUERY PLAN
------------------------------------------------------------------------------------------
XN Merge (cost=1000000000136.38..1000000000137.82 rows=576 width=17)
Merge Key: eventname
-> XN Network (cost=1000000000136.38..1000000000137.82 rows=576
width=17)
Send to leader
-> XN Sort (cost=1000000000136.38..1000000000137.82 rows=576
width=17)
Sort Key: eventname
```

```
-> XN Unique (cost=0.00..109.98 rows=576 width=17)
-> XN Seq Scan on event (cost=0.00..87.98 rows=8798
width=17)
(8 rows)

select * from stl_plan_info where query=240 order by nodeid desc;

query | nodeid | segment | step | locus | plannode | startupcost |
totalcost | rows | bytes
-------+--------+---------+------+-------+----------+-----------------
+-----------------+------+--------
240 | 5 | 0 | 0 | 0 | 104 | 0                | 87.98   | 8798 | 149566
240 | 5 | 0 | 1 | 0 | 104 | 0                | 87.98   | 8798 | 149566
240 | 4 | 0 | 2 | 0 | 117 | 0                | 109.975 | 576  | 9792
240 | 4 | 0 | 3 | 0 | 117 | 0                | 109.975 | 576  | 9792
240 | 4 | 1 | 0 | 0 | 117 | 0                | 109.975 | 576  | 9792
240 | 4 | 1 | 1 | 0 | 117 | 0                | 109.975 | 576  | 9792
240 | 3 | 1 | 2 | 0 | 114 | 1000000000136.38 | 1000000000137.82 | 576 | 9792
240 | 3 | 2 | 0 | 0 | 114 | 1000000000136.38 | 1000000000137.82 | 576 | 9792
240 | 2 | 2 | 1 | 0 | 123 | 1000000000136.38 | 1000000000137.82 | 576 | 9792
240 | 1 | 3 | 0 | 0 | 122 | 1000000000136.38 | 1000000000137.82 | 576 | 9792
(10 rows)
```

STL_PROJECT

Contiene righe per le fasi di query utilizzate per valutare delle espressioni.

STL_PROJECT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_PROJECT contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
checksum	bigint	Queste informazioni sono solo per uso interno.

Query di esempio

L'esempio seguente restituisce tutte le righe per le fasi di query utilizzate per valutare espressioni per la sezione 0 e il segmento 1.

```
select query, step, starttime, endtime, tasknum, rows
from stl_project
where slice=0 and segment=1;
```

```
 query  | step |         starttime   |          endtime    | tasknum | rows
--------+------+---------------------+---------------------+---------+------
  86399 |    2 | 2013-08-29 22:01:21 | 2013-08-29 22:01:21 |      25 |   -1
  86399 |    3 | 2013-08-29 22:01:21 | 2013-08-29 22:01:21 |      25 |   -1
    719 |    1 | 2013-08-12 22:38:33 | 2013-08-12 22:38:33 |       7 |   -1
  86383 |    1 | 2013-08-29 21:58:35 | 2013-08-29 21:58:35 |       7 |   -1
    714 |    1 | 2013-08-12 22:38:17 | 2013-08-12 22:38:17 |       2 |   -1
  86375 |    1 | 2013-08-29 21:57:59 | 2013-08-29 21:57:59 |       2 |   -1
  86397 |    2 | 2013-08-29 22:01:20 | 2013-08-29 22:01:20 |      19 |   -1
    627 |    1 | 2013-08-12 22:34:13 | 2013-08-12 22:34:13 |      34 |   -1
  86326 |    2 | 2013-08-29 21:45:28 | 2013-08-29 21:45:28 |      34 |   -1
  86326 |    3 | 2013-08-29 21:45:28 | 2013-08-29 21:45:28 |      34 |   -1
  86325 |    2 | 2013-08-29 21:45:27 | 2013-08-29 21:45:27 |      28 |   -1
  86371 |    1 | 2013-08-29 21:57:42 | 2013-08-29 21:57:42 |       4 |   -1
 111100 |    2 | 2013-09-03 19:04:45 | 2013-09-03 19:04:45 |      12 |   -1
    704 |    2 | 2013-08-12 22:36:34 | 2013-08-12 22:36:34 |      37 |   -1
    649 |    2 | 2013-08-12 22:34:47 | 2013-08-12 22:34:47 |      38 |   -1
    649 |    3 | 2013-08-12 22:34:47 | 2013-08-12 22:34:47 |      38 |   -1
    632 |    2 | 2013-08-12 22:34:22 | 2013-08-12 22:34:22 |      13 |   -1
    705 |    2 | 2013-08-12 22:36:48 | 2013-08-12 22:36:49 |      13 |   -1
    705 |    3 | 2013-08-12 22:36:48 | 2013-08-12 22:36:49 |      13 |   -1
      3 |    1 | 2013-08-12 20:07:40 | 2013-08-12 20:07:40 |       3 |   -1
  86373 |    1 | 2013-08-29 21:57:58 | 2013-08-29 21:57:58 |       3 |   -1
 107976 |    1 | 2013-09-03 04:05:12 | 2013-09-03 04:05:12 |       3 |   -1
  86381 |    1 | 2013-08-29 21:58:35 | 2013-08-29 21:58:35 |       8 |   -1
  86396 |    1 | 2013-08-29 22:01:20 | 2013-08-29 22:01:20 |      15 |   -1
    711 |    1 | 2013-08-12 22:37:10 | 2013-08-12 22:37:10 |      20 |   -1
  86324 |    1 | 2013-08-29 21:45:27 | 2013-08-29 21:45:27 |      24 |   -1
(26 rows)
```

STL_QUERY

Restituisce informazioni di esecuzione su una query del database.

> **Note**
>
> Le viste STL_QUERY e STL_QUERYTEXT contengono solo informazioni sulle query, non su altre utility o comandi DDL. Per un elenco e le informazioni su tutte le istruzioni

eseguite da Amazon Redshift, è possibile anche eseguire query sulle tabelle STL_DDLTEXT e STL_UTILITYTEXT. Per un elenco completo di tutte le istruzioni eseguite da Amazon Redshift, è possibile eseguire una query sulla vista SVL_STATEMENTTEXT.

STL_QUERY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etichetta definita con un comando SET QUERY GROUP. Se la query non è basata su file o non è impostato il parametro QUERY_GROUP, questo valore del campo è `default`.
xid	bigint	ID transazione.
pid	integer	ID processo. In genere, tutte le query in una sessione sono eseguite nello stesso processo, quindi questo valore di solito rimane costante se esegui una serie di query nella stessa sessione. Seguendo determinati eventi interni, Amazon Redshift può riavviare una sessione attiva e assegnare un nuovo PID. Per ulteriori informazioni, consulta STL_RESTARTED_SESSIONS.

Nome colonna	Tipo di dati	Descrizione
database	character(32)	Il nome del database al quale l'utente era collegato al momento del rilascio della query.
querytxt	character(4000)	Testo reale di query per la query.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precision e a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precision e a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
aborted	integer	Se la query è stata interrotta dal sistema o annullata dall'utente, questa colonna contiene **1**. Se la query è stata eseguita fino a completamento (inclusa la restituzi one dei risultati al client), questa colonna contiene **`0`**. Se un client si disconnette prima di ricevere i risultati, la query sarà contrassegnata come annullata (**1**), anche se il completamento è riuscito nel back-end.
insert_pristine	integer	Indica se l'esecuzione delle query di scrittura è/era possibile quando la query corrente è/era in esecuzione. 1 = nessuna query di scrittura consentita. 0 = query di scrittura consentite. Questa colonna è da utilizzarsi per scopi di debug.

Nome colonna	Tipo di dati	Descrizione
concurrency_scaling_status	integer	Indica se la query è stata eseguita nel cluster principale o in un cluster di dimensionamento della concorrenza. I valori possibili sono i seguenti: 0 - Eseguita nel cluster principale 1 - Eseguita in un cluster con dimensionamento simultaneo Superiore a 1 - Eseguita nel cluster principale

Query di esempio

La query seguente elenca le cinque query più recenti.

```
select query, trim(querytxt) as sqlquery
from stl_query
order by query desc limit 5;

query |                                  sqlquery
------+-------------------------------------------------
129 | select query, trim(querytxt) from stl_query order by query;
128 | select node from stv_disk_read_speeds;
127 | select system_status from stv_gui_status
126 | select * from systable_topology order by slice
125 | load global dict registry
(5 rows)
```

La query seguente restituisce il tempo trascorso in ordine decrescente per le query eseguite il 15 febbraio 2013.

```
select query, datediff(seconds, starttime, endtime),
trim(querytxt) as sqlquery
from stl_query
where starttime >= '2013-02-15 00:00' and endtime < '2013-02-16 00:00'
order by date_diff desc;

 query | date_diff |  sqlquery
```

```
-------+-----------+-----------------------------------------
 55    |       119 | padb_fetch_sample: select count(*) from category
121    |         9 | select * from svl_query_summary;
181    |         6 | select * from svl_query_summary where query in(179,178);
172    |         5 | select * from svl_query_summary where query=148;
...
(189 rows)
```

La seguente query mostra il tempo di attesa in coda e il tempo di esecuzione delle query. Le query con `concurrency_scaling_status = 1` sono state eseguite in un cluster di dimensionamento della concorrenza. Tutte le altre query sono state eseguite nel cluster principale.

```
SELECT w.service_class AS queue
     , q.concurrency_scaling_status
     , COUNT( * ) AS queries
     , SUM( q.aborted )  AS aborted
     , SUM( ROUND( total_queue_time::NUMERIC / 1000000,2 ) ) AS queue_secs
     , SUM( ROUND( total_exec_time::NUMERIC / 1000000,2 ) )  AS exec_secs
FROM stl_query q
     JOIN stl_wlm_query w
          USING (userid,query)
WHERE q.userid > 1
  AND service_class > 5
  AND q.starttime > '2019-03-01 16:38:00'
  AND q.endtime   < '2019-03-01 17:40:00'
GROUP BY 1,2
ORDER BY 1,2;
```

STL_QUERY_METRICS

Contiene informazioni di parametro, come il numero di righe elaborate, l'uso della CPU, input/output e l'uso del disco per le query che hanno completato l'esecuzione in code di query definite dall'utente (classi di servizio). Per visualizzare i parametri per le query attive correntemente in esecuzione, consultare la vista di sistema STV_QUERY_METRICS.

I parametri delle query sono campionati a intervalli di un secondo. Di conseguenza, differenti esecuzioni della stessa query potrebbero restituire orari leggermente differenti. Inoltre, i segmenti di query che vengono eseguiti in meno di un secondo potrebbero non essere registrati.

STL_QUERY_METRICS traccia e aggrega i parametri a livello di query, di segmento e di fase. Per ulteriori informazioni sui segmenti e sulle fasi di query, consultare Pianificazione di query e flusso di

lavoro di esecuzione. Molti parametri (come `max_rows`, `cpu_time` e così via) sono sommati sulle sezioni di nodo. Per ulteriori informazioni sulle sezioni di nodo, consultare Architettura del sistema di data warehouse.

Per determinare il livello in cui la riga fornisce i parametri, esamina le colonne `segment` e `step_type`.

- Se entrambe le colonne `segment` e `step_type` sono `-1`, la riga fornisce parametri a livello di query.
- Se `segment` non è `-1` e `step_type` è `-1`, la riga fornisce parametri a livello di segmento.
- Se entrambe le colonne `segment` e `step_type` non sono `-1`, la riga fornisce parametri a livello di fase.

La vista SVL_QUERY_METRICS e la vista SVL_QUERY_METRICS_SUMMARY aggregano i dati in questa vista e presentano le informazioni in un modulo più accessibile.

STL_QUERY_METRICS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha eseguito la query che ha generato la voce.
service_class	integer	ID per la classe di servizio. Le code di query sono definite nella configurazione WLM. I parametri sono restituiti solo per le code definite dall'utente.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.

Nome colonna	Tipo di dati	Descrizione
segment	integer	Numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. I segmenti di query possono essere eseguiti in parallelo. Ogni segmento viene eseguito in un singolo processo. Se il valore del segmento è -1, viene eseguito il rollup dei valori di segmento dei parametri a livello della query.
step_type	integer	Tipo di fase eseguita. Per una descrizione dei tipi di fase, consultare Tipo di fase.
starttime	timestamp	Orario in UTC in cui la query ha avviato l'esecuzione, con 6 cifre di precisione per le frazioni di secondo. Ad esempio: `2009-06-12 11:29:19.131358` .
slices	integer	Numero di sezioni per il cluster.
max_rows	bigint	Numero massimo di righe prodotte per una fase, aggregate tra tutte le sezioni.
righe	bigint	Numero di righe elaborate da una fase.
max_cpu_time	bigint	Tempo CPU massimo utilizzato in microsecondi. A livello del segmento, il tempo CPU massimo utilizzato dal segmento tra tutte le sezioni. A livello della query, il tempo CPU massimo utilizzato dal qualsiasi segmento di query.
cpu_time	bigint	Tempo CPU utilizzato in microsecondi. A livello del segmento, il tempo CPU totale per il segmento tra tutte le sezioni. A livello della query, la somma del tempo CPU per la query tra tutte le sezioni e tutti i segmenti.
max_block s_read	bigint	Numero massimo di blocchi da 1 MB letti dal segmento, aggregati tra tutte le sezioni. A livello del segmento, il numero massimo di blocchi da 1 MB letti per il segmento tra tutte le sezioni. A livello della query, il numero massimo di blocchi da 1 MB letti da qualsiasi segmento di query.

Nome colonna	Tipo di dati	Descrizione
blocks_read	bigint	Numero di blocchi da 1 MB letti dalla query o dal segmento.
max_run_time	bigint	Tempo massimo trascorso per un segmento, in microsecondi. A livello del segmento, il tempo di esecuzione massimo per il segmento tra tutte le sezioni. A livello della query, il tempo di esecuzione per qualsiasi segmento di query.
run_time	bigint	Tempo di esecuzione totale, sommato tra le sezioni. Il tempo di esecuzione non include il tempo di attesa. A livello del segmento, il tempo di esecuzione per il segmento, sommato tra tutte le sezioni. A livello della query, il tempo di esecuzione per la query sommato tra tutte le sezioni e tutti i segmenti. Poiché questo valore è una somma, il tempo di esecuzione non è correlato al tempo di esecuzione della query.
max_block s_to_disk	bigint	La quantità massima di spazio su disco utilizzata per scrivere risultati intermedi, in blocchi di MB. A livello del segmento, la quantità massima di spazio su disco utilizzato dal segmento tra tutte le sezioni. A livello della query, la quantità massima di spazio su disco utilizzato da qualsiasi segmento di query.
blocks_to_disk	bigint	La quantità di spazio su disco utilizzata da una query o da un segmento per scrivere risultati intermedi, in blocchi di MB.
step	integer	La fase di query eseguita.
max_query _scan_size	bigint	La dimensione massima di dati sottoposti a scansione da una query in MB. A livello del segmento, la quantità massima di dati sottoposti a scansione dal segmento tra tutte le sezioni. A livello della query, la quantità massima di dati sottoposti a scansione da qualsiasi segmento di query.
query_scan_size	bigint	La dimensione dei dati sottoposti a scansione da una query in MB.

Nome colonna	Tipo di dati	Descrizione
query_priority	integer	La priorità della query. I valori possibili sono -1, 0, 1, 2, 3 e 4, dove -1 indica che la priorità della query non è supportata.
query_queue_time	bigint	La quantità di tempo espressa in microsecondi di permanenza della query nella coda.
service_class_name	character (64)	Il nome della classe dei servizi.

Query di esempio

Per trovare le query con un tempo di CPU elevato (più di 1.000 secondi), esegui la seguente query.

```
Select query, cpu_time / 1000000 as cpu_seconds
from stl_query_metrics where segment = -1 and cpu_time > 1000000000
order by cpu_time;

query | cpu_seconds
------+------------
25775 |        9540
```

Per trovare query attive con un nested loop join che hanno restituito più di un milione di righe, esegui la query seguente.

```
select query, rows
from stl_query_metrics
where step_type = 15 and rows > 1000000
order by rows;

query | rows
------+-----------
25775 | 2621562702
```

Per trovare query attive eseguite per più di 60 secondi e con tempo CPU utilizzato inferiore a 10 secondi, esegui la query seguente.

```
select query, run_time/1000000 as run_time_seconds
from stl_query_metrics
```

```
where segment = -1 and run_time > 60000000 and cpu_time < 10000000;

query | run_time_seconds
------+-----------------
25775 |              114
```

STL_QUERYTEXT

Acquisisce il testo di query per i comandi SQL.

Eseguire una query sulla tabella STL_QUERYTEXT per acquisire l'SQL registrato per le seguenti istruzioni:

- SELECT, SELECT INTO
- INSERT, UPDATE, DELETE
- COPY
- UNLOAD
- Le query generate dall'esecuzione di VACUUM e ANALYZE
- CREATE TABLE AS (CTAS)

Per eseguire una query dell'attività per queste istruzioni in un dato periodo di tempo, unisci le viste STL_QUERYTEXT e STL_QUERY.

> **Note**
>
> Le viste STL_QUERY e STL_QUERYTEXT contengono solo informazioni sulle query, non su altre utility o comandi DDL. Per un elenco e le informazioni su tutte le istruzioni eseguite da Amazon Redshift, è possibile anche eseguire query sulle tabelle STL_DDLTEXT e STL_UTILITYTEXT. Per un elenco completo di tutte le istruzioni eseguite da Amazon Redshift, è possibile eseguire una query sulla vista SVL_STATEMENTTEXT.

Consulta anche STL_DDLTEXT, STL_UTILITYTEXT e SVL_STATEMENTTEXT.

STL_QUERYTEXT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_TEXT. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
xid	bigint	ID transazione.
pid	integer	ID processo. In genere, tutte le query in una sessione sono eseguite nello stesso processo, quindi questo valore di solito rimane costante se esegui una serie di query nella stessa sessione. Seguendo determinati eventi interni, Amazon Redshift può riavviare una sessione attiva e assegnare un nuovo PID. Per ulteriori informazioni, consulta STL_RESTARTED_SESSIONS. È possibile utilizzare questa colonna per eseguire la connessione alla vista STL_ERROR.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sequenza	integer	Quando una singola istruzione contiene più di 200 caratteri , vengono registrate delle righe aggiuntive per tale istruzion e. La sequenza 0 è la prima riga, 1 la seconda e così via.
text	character(200)	Testo SQL, in incrementi da 200 caratteri. Questo campo potrebbe contenere caratteri speciali come barra rovesciat a (\\) e nuova riga (\n).

Query di esempio

È possibile utilizzare la funzione PG_BACKEND_PID() per recuperare le informazioni sulla sessione corrente. Ad esempio, la query seguente restituisce l'ID di query e una porzione del testo di query delle query completate nella sessione corrente.

```
select query, substring(text,1,60)
from stl_querytext
where pid = pg_backend_pid()
order by query desc;

 query |                          substring
-------+------------------------------------------------------------
 28262 | select query, substring(text,1,80) from stl_querytext where
 28252 | select query, substring(path,0,80) as path from stl_unload_l
 28248 | copy category from 's3://dw-tickit/manifest/category/1030_ma
 28247 | Count rows in target table
 28245 | unload ('select * from category') to 's3://dw-tickit/manifes
 28240 | select query, substring(text,1,40) from stl_querytext where
(6 rows)
```

Ricostruzione dell'SDL archiviato

Per ricostruire l'SQL archiviato nella colonna `text` di STL_QUERYTEXT, eseguire un'istruzione SELECT per creare un SQL da 1 o più parti nella colonna `text`. Prima di eseguire l'SQL ricostruito, sostituire un carattere speciale qualsiasi (\n) con una nuova riga. Il risultato della dichiarazione SELECT seguente è dato da una riga di SQL ricostruiti nel campo `query_statement`.

```
SELECT query, LISTAGG(CASE WHEN LEN(RTRIM(text)) = 0 THEN text ELSE RTRIM(text) END)
 WITHIN GROUP (ORDER BY sequence) as query_statement, COUNT(*) as row_count
FROM stl_querytext GROUP BY query ORDER BY query desc;
```

Ad esempio, la query seguente seleziona 3 colonne. La query stessa è più lunga di 200 caratteri ed è archiviata in parti in STL_QUERYTEXT.

```
select
1 AS a01234567890123456789012345678901234567890123456789012345678901234567890,
2 AS b01234567890123456789012345678901234567890123456789012345678901234567890,
3 AS b01234567890123456789012345678901234567890123456789012345678901234
FROM stl_querytext;
```

In questo esempio, la query è archiviata in 2 parti (righe) nella colonna `text` di STL_QUERYTEXT.

```
select query, sequence, text
from stl_querytext where query=pg_last_query_id() order by query desc, sequence limit
 10;
```

```
query  | sequence |
                         text

-------+----------
+-----------------------------------------------------------------------------------------------------
    45 |        0 | select\n1 AS
 a012345678901234567890123456789012345678901234567890123456789012345678901234567890,\n2 AS
 b012345678901234567890123456789012345678901234567890123456789012345678901234567890,\n3 AS
 b012345678901234567890123456789012345678901234
    45 |        1 | \nFROM stl_querytext;
```

Per ricostruire l'SQL archiviato in STL_QUERYTEXT, eseguire l'SQL seguente.

```
select LISTAGG(CASE WHEN LEN(RTRIM(text)) = 0 THEN text ELSE RTRIM(text) END, '')
 within group (order by sequence) AS text
from stl_querytext where query=pg_last_query_id();
```

Per utilizzare l'SQL ricostruito derivante nel client, sostituire qualsiasi carattere speciale (\n) con una nuova riga.

```
                    text

-----------------------------------------------------------------------------------------------------
 select\n1 AS a012345678901234567890123456789012345678901234567890123456789012345678901234567890,
\n2 AS b012345678901234567890123456789012345678901234567890123456789012345678901234567890,\n3 AS
 b012345678901234567890123456789012345678901234\nFROM stl_querytext;
```

STL_REPLACEMENTS

Mostra un log che registra il momento in cui dei caratteri UTF-8 non validi sono stati sostituiti dal comando COPY con l'opzione ACCEPTINVCHARS. Una voce di log viene aggiunta a

STL_REPLACEMENTS per ognuna delle prime 100 righe su ogni sezione di nodo che ha richiesto almeno una sostituzione.

STL_REPLACEMENTS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_NESTLOOP contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_COPY_REPLACEMENTS. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero della sezione di nodo in cui si è verificata la sostituzione.
tbl	integer	ID tabella.
starttime	timestamp	Ora di inizio in UTC del comando COPY.
session	integer	ID di sessione per la sessione che esegue il comando COPY.
filename	character (256)	Percorso completo verso il file di input per il comando COPY.
line_number	bigint	Numero di riga nel file di dati di input che conteneva un carattere UTF-8 non valido. Un -1 indica che il numero di riga

Nome colonna	Tipo di dati	Descrizione
		non è disponibile, ad esempio quando si copia da un file di dati colonnare.
colname	character (127)	Primo campo che conteneva un carattere UTF-8 non valido.
raw_line	character (1024)	Dati di caricamento non elaborati che contenevano un carattere UTF-8 non valido.

Query di esempio

L'esempio seguente restituisce le sostituzioni per l'operazione COPY più recente.

```
select query, session, filename, line_number, colname
from stl_replacements
where query = pg_last_copy_id();

 query | session |   filename                          | line_number | colname
 ------+---------+-------------------------------------+-------------+--------
    96 |    6314 | s3://mybucket/allusers_pipe.txt   |         251 | city
    96 |    6314 | s3://mybucket/allusers_pipe.txt   |         317 | city
    96 |    6314 | s3://mybucket/allusers_pipe.txt   |         569 | city
    96 |    6314 | s3://mybucket/allusers_pipe.txt   |         623 | city
    96 |    6314 | s3://mybucket/allusers_pipe.txt   |         694 | city
...
```

STL_RESTARTED_SESSIONS

Per mantenere una disponibilità continua in seguito a determinati eventi interni, Amazon Redshift potrebbe riavviare una sessione attiva con un nuovo ID di processo (PID). Quando Amazon Redshift riavvia una sessione, STL_RESTARTED_SESSIONS registra il nuovo PID e il vecchio PID.

Per maggiori informazioni, consultare gli esempi seguenti in questa sezione.

STL_RESTARTED_SESSIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_SESSION_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
currenttime	timestamp	Ora dell'evento.
dbname	character (50)	Nome del database associato alla sessione.
newpid	integer	ID di processo per la sessione riavviata.
oldpid	integer	ID di processo per la sessione originale.
username	character (50)	Nome dell'utente associato alla sessione.
remotehost	character (45)	Nome o indirizzo IP dell'host remoto.
remoteport	character (32)	Numero di porta dell'host remoto.
parkedtime	timestamp	Queste informazioni sono solo per uso interno.
session_vars	character (2000)	Queste informazioni sono solo per uso interno.

Query di esempio

L'esempio seguente collega STL_RESTARTED_SESSIONS a STL_SESSIONS per mostrare i nomi utente per le sessioni che sono state riavviate.

```
select process, stl_restarted_sessions.newpid, user_name
from stl_sessions
```

```
inner join stl_restarted_sessions on stl_sessions.process =
 stl_restarted_sessions.oldpid
order by process;

...
```

STL_RETURN

Contiene dettagli per le fasi return nelle query. Una fase return restituisce i risultati di query completate sui nodi di calcolo al nodo principale. Il nodo principale unisce i dati e restituisce i risultati al client che li richiede. Per le query completate sul nodo principale, una fase return restituisce i risultati al client.

Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. Per ulteriori informazioni, consulta Elaborazione query.

STL_RETURN è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_RETURN contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.

Nome colonna	Tipo di dati	Descrizione
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
byte	bigint	Dimensione, in byte, di tutte le righe di output della fase.
packets	integer	Numero complessivo di pacchetti inviati nella rete.
checksum	bigint	Queste informazioni sono solo per uso interno.

Query di esempio

La seguente query mostra quali fasi sono state eseguite in ogni sezione nella query più recente.

```
SELECT query, slice, segment, step, endtime, rows, packets
from stl_return where query = pg_last_query_id();

 query |  slice | segment | step |          endtime           | rows | packets
-------+--------+---------+------+----------------------------+------+---------
     4 |      2 |       3 |    2 | 2013-12-27 01:43:21.469043 |    3 |       0
```

```
     4 |      3 |       3 |    2 | 2013-12-27 01:43:21.473321 |    0 |       0
     4 |      0 |       3 |    2 | 2013-12-27 01:43:21.469118 |    2 |       0
     4 |      1 |       3 |    2 | 2013-12-27 01:43:21.474196 |    0 |       0
     4 |      4 |       3 |    2 | 2013-12-27 01:43:21.47704  |    2 |       0
     4 |      5 |       3 |    2 | 2013-12-27 01:43:21.478593 |    0 |       0
     4 |  12811|       4 |    1 | 2013-12-27 01:43:21.480755 |    0 |       0
(7 rows)
```

STL_S3CLIENT

Registra il tempo di trasferimento e altri parametri relativi alle prestazioni.

Utilizza la tabella STL_S3CLIENT per sapere quanto tempo è stato impiegato per trasferire i dati da Amazon S3.

STL_S3CLIENT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
recordtime	timestamp	Ora in cui il record è stato registrato.
pid	integer	ID processo. Tutte le query in una sessione sono eseguite nello stesso processo, quindi questo valore rimane costante se si esegue una serie di query nella stessa sessione.
http_method	character (64)	Nome del metodo HTTP corrispondente alla richiesta di Amazon S3.

Nome colonna	Tipo di dati	Descrizione
bucket	character (64)	Nome bucket S3.
Chiave	character (256)	Chiave corrispondente all'oggetto Amazon S3
transfer_size	bigint	Numero di byte trasferiti.
data_size	bigint	Numero di byte di dati. Questo valore è uguale a transfer_size per i dati non compressi. Se è stata utilizzata la compressione, questo equivale alla dimensione dei dati non compressi.
start_time	bigint	Ora di inizio del trasferimento (in microsecondi dal 1° gennaio 2000).
end_time	bigint	Ora di fine del trasferimento (in microsecondi dal 1° gennaio del 2000).
transfer_time	bigint	Tempo impiegato per il trasferimento (in microsecondi).
compression_time	bigint	Porzione del tempo di trasferimento impiegata a decomprim ere i dati (in microsecondi).
connect_time	bigint	Tempo trascorso dall'avvio al completamento della connessio ne al server remoto (in microsecondi).
app_connect_time	bigint	Tempo trascorso dall'avvio al completamento della connessio ne/handshake SSL con l'host remoto (in microsecondi).
retries	bigint	Numero di volte in cui il trasferimento è stato ripetuto.
request_id	char(32)	ID di richiesta dall'intestazione della risposta HTTP di Amazon S3
extended_ request_id	char(128)	ID di richiesta esteso dalla risposta dell'intestazione HTTP Amazon S3 (x-amz-id-2).
ip_address	char(64)	Indirizzo IP del server (ip V4 o V6).

Nome colonna	Tipo di dati	Descrizione
is_partial	integer	Valore che se true (1) indica che il file di input viene diviso in intervalli durante un'operazione COPY. Se il valore è false (0), il file di input non viene diviso.
start_offset	bigint	Valore che, se il file di input viene diviso durante un'operaz ione COPY, indica il valore di offset della divisione (in byte). Se il file non è diviso, questo valore è 0.

Query di esempio

La seguente query restituisce il tempo impiegato per caricare i file utilizzando un comando COPY.

```
select slice, key, transfer_time
from stl_s3client
where query = pg_last_copy_id();
```

Risultato

```
 slice |   key                         | transfer_time
 ------+-------------------------------+---------------
     0 | listing10M0003_part_00        |    16626716
     1 | listing10M0001_part_00        |    12894494
     2 | listing10M0002_part_00        |    14320978
     3 | listing10M0000_part_00        |    11293439
  3371 | prefix=listing10M;marker=     |       99395
```

La query seguente converte `start_time` e `end_time` in un timestamp.

```
select userid,query,slice,pid,recordtime,start_time,end_time,
'2000-01-01'::timestamp + (start_time/1000000.0)* interval '1 second' as start_ts,
'2000-01-01'::timestamp + (end_time/1000000.0)* interval '1 second' as end_ts
from stl_s3client where query> -1 limit 5;
```

```
 userid | query | slice |  pid  |         recordtime         |   start_time    |
 end_time     |          start_ts          |           end_ts
--------+-------+-------+-------+----------------------------+-----------------
+-----------------+----------------------------+----------------------------
```

```
     0 |     0 |     0 | 23449 | 2019-07-14 16:27:17.207839 | 616436837154256 |
616436837207838 | 2019-07-14 16:27:17.154256 | 2019-07-14 16:27:17.207838
     0 |     0 |     0 | 23449 | 2019-07-14 16:27:17.252521 | 616436837208208 |
616436837252520 | 2019-07-14 16:27:17.208208 | 2019-07-14 16:27:17.25252
     0 |     0 |     0 | 23449 | 2019-07-14 16:27:17.284376 | 616436837208460 |
616436837284374 | 2019-07-14 16:27:17.20846  | 2019-07-14 16:27:17.284374
     0 |     0 |     0 | 23449 | 2019-07-14 16:27:17.285307 | 616436837208980 |
616436837285306 | 2019-07-14 16:27:17.20898  | 2019-07-14 16:27:17.285306
     0 |     0 |     0 | 23449 | 2019-07-14 16:27:17.353853 | 616436837302216 |
616436837353851 | 2019-07-14 16:27:17.302216 | 2019-07-14 16:27:17.353851
```

STL_S3CLIENT_ERROR

Registra gli errori prodotti da una sezione durante il caricamento di un file da Amazon S3.

Utilizza la STL_S3CLIENT_ERROR per trovare i dettagli degli errori prodotti durante il trasferimento dei dati da Amazon S3 come parte di un comando COPY.

STL_S3CLIENT_ERROR è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema. L'ID di query -1 è per uso interno.
sliceid	integer	Numero che identifica la sezione in cui è stata eseguita la query.
recordtime	timestamp	Ora in cui il record è stato registrato.

Nome colonna	Tipo di dati	Descrizione
pid	integer	ID processo. Tutte le query in una sessione sono eseguite nello stesso processo, quindi questo valore rimane costante se si esegue una serie di query nella stessa sessione.
http_method	character (64)	Nome del metodo HTTP corrispondente alla richiesta di Amazon S3.
bucket	character (64)	Nome del bucket Amazon S3.
Chiave	character (256)	Chiave corrispondente all'oggetto Amazon S3
error	character (1024)	Messaggio di errore.
is_partial	integer	Valore che, se true (1), indica che il file di input viene diviso in intervalli durante un'operazione COPY. Se il valore è false (0), il file di input non viene diviso.
start_offset	bigint	Valore che, se il file di input viene diviso durante un'operaz ione COPY, indica il valore di offset della divisione (in byte). Se il file non è diviso, questo valore è 0.

Note per l'utilizzo

Se noti più errori con "Connessione scaduta", potrebbe esserci un problema di reti. Se stai utilizzando un Routing VPC avanzato, verifica di avere un percorso di rete valido tra il VPC di cluster e le risorse di dati. Per ulteriori informazioni, consultare Routing VPC avanzato di Amazon Redshift.

Query di esempio

La seguente query restituisce gli errori dai comandi COPY completati durante la sessione corrente.

```
select query, sliceid, substring(key from 1 for 20) as file,
substring(error from 1 for 35) as error
from stl_s3client_error
where pid = pg_backend_pid()
```

```
order by query desc;
```

Risultato

```
 query  | sliceid |     file           |              error

--------+---------+--------------------+-----------------------------------
 362228 |      12 | part.tbl.25.159.gz | transfer closed with 1947655 bytes
 362228 |      24 | part.tbl.15.577.gz | transfer closed with 1881910 bytes
 362228 |       7 | part.tbl.22.600.gz | transfer closed with 700143 bytes r
 362228 |      22 | part.tbl.3.34.gz   | transfer closed with 2334528 bytes
 362228 |      11 | part.tbl.30.274.gz | transfer closed with 699031 bytes r
 362228 |      30 | part.tbl.5.509.gz  | Unknown SSL protocol error in conne
 361999 |      10 | part.tbl.23.305.gz | transfer closed with 698959 bytes r
 361999 |      19 | part.tbl.26.582.gz | transfer closed with 1881458 bytes
 361999 |       4 | part.tbl.15.629.gz | transfer closed with 2275907 bytes
 361999 |      20 | part.tbl.6.456.gz  | transfer closed with 692162 bytes r
(10 rows)
```

STL_SAVE

Contiene dettagli per le fasi di salvataggio nelle query. Una fase di salvataggio salva il flusso di input in una tabella transitoria. Una tabella transitoria è una tabella temporanea che archivia i risultati intermedi durante l'esecuzione della query.

Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. Per ulteriori informazioni, consulta Elaborazione query.

STL_SAVE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_SAVE contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
byte	bigint	Dimensione, in byte, di tutte le righe di output della fase.
tbl	integer	ID della tabella transitoria materializzata.
is_diskbased	character(1)	Se questa fase della query è stata eseguita come operazione basata su disco: true (`t`) o false (`f`).
workmem	bigint	Numero di byte della memoria di lavoro assegnati alla fase.

Query di esempio

La seguente query mostra quali fasi sono state salvate in ogni sezione nella query più recente.

```
select query, slice, segment, step, tasknum, rows,  tbl
from stl_save where query = pg_last_query_id();

 query | slice | segment | step | tasknum | rows | tbl
-------+-------+---------+------+---------+------+-----
 52236 |     3 |       0 |    2 |      21 |    0 | 239
 52236 |     2 |       0 |    2 |      20 |    0 | 239
 52236 |     2 |       2 |    2 |      20 |    0 | 239
 52236 |     3 |       2 |    2 |      21 |    0 | 239
 52236 |     1 |       0 |    2 |      21 |    0 | 239
 52236 |     0 |       0 |    2 |      20 |    0 | 239
 52236 |     0 |       2 |    2 |      20 |    0 | 239
 52236 |     1 |       2 |    2 |      21 |    0 | 239
(8 rows)
```

STL_SCAN

Analizza le fasi di scansione di tabella per le query. Il numero di fase per le righe in questa tabella è sempre 0 perché una scansione è la prima fase in un segmento.

STL_SCAN è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_SCAN contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
byte	bigint	Dimensione, in byte, di tutte le righe di output della fase.
fetches	bigint	Queste informazioni sono solo per uso interno.
tipo	integer	ID del tipo di scansione. Per un elenco di valori validi, consultare la tabella seguente.
tbl	integer	ID tabella.

Nome colonna	Tipo di dati	Descrizione
is_rrscan	character(1)	Se true (t), indica che in questa fase è stata utilizzata la scansione a intervallo limitato.
is_delayed_scan	character(1)	Queste informazioni sono solo per uso interno.
rows_pre_filter	bigint	Per le scansioni di tabelle permanenti, il numero totale di righe emesse prima di aver applicato filtri alle righe contrassegnate per l'eliminazione (righe fantasma) e prima dell'applicazione di filtri di query definiti dall'utente.
rows_pre_user_filter	bigint	Per le scansioni di tabelle permanenti, il numero di righe elaborate dopo aver applicato filtri alle righe contrassegnate per l'eliminazione (righe fantasma) ma prima dell'applicazione di filtri di query definiti dall'utente.
perm_table_name	character(136)	Per le scansioni di tabelle permanenti, il nome della tabella su cui è eseguita la scansione.
is_rlf_scan	character(1)	Se true (t), indica che nella fase è stata utilizzata l'applicazione di filtri a livello di riga.
is_rlf_scan_reason	integer	Queste informazioni sono solo per uso interno.
num_em_blocks	integer	Queste informazioni sono solo per uso interno.
checksum	bigint	Queste informazioni sono solo per uso interno.
runtime_filtering	character(1)	Se true (t), indica che vengono applicati filtri di runtime.
scan_region	integer	Queste informazioni sono solo per uso interno.

Nome colonna	Tipo di dati	Descrizione
num_sortkey_as_predicate	integer	Queste informazioni sono solo per uso interno.
row_fetcher_state	integer	Queste informazioni sono solo per uso interno.
consumed_scan_ranges	bigint	Queste informazioni sono solo per uso interno.
work_stealing_reason	bigint	Queste informazioni sono solo per uso interno.
is_vectorized_scan	character(1)	Queste informazioni sono solo per uso interno.
is_vectorized_scan_reason	integer	Queste informazioni sono solo per uso interno.
row_fetcher_reason	bigint	Queste informazioni sono solo per uso interno.
topology_signature	bigint	Queste informazioni sono solo per uso interno.
use_tpm_partition	character(1)	Queste informazioni sono solo per uso interno.
is_rrscan_expr	character(1)	Queste informazioni sono solo per uso interno.

Nome colonna	Tipo di dati	Descrizione
scanned_mega_value	character(1)	Queste informazioni sono solo per uso interno. Queste informazioni mostrano se la fase di scansione specificata ha scansionato un valore elevato. Un valore elevato verrà archiviato in più blocchi. La dimensione del blocco è 1 MB per impostazione predefinita, un valore elevato è superiore a 1 MB in un'impostazione predefinita.

Tipi di scansione

ID di tipo	Descrizione
1	Dati dalla rete.
2	Tabelle d'utente permanenti nella memoria condivisa compressa.
3	Tabelle transitorie in linea.
21	Carica file da Amazon S3.
22	Carica tabelle da Amazon DynamoDB.
23	Carica i dati da una connessione SSH remota.
24	Carica i dati da un cluster remoto (regione ordinata). È utilizzato per il ridimensionamento.
25	Carica i dati da un cluster remoto (regione non ordinata). È utilizzato per il ridimensionamento.
28	Legge i dati da una vista di serie temporali con UNION ALL su più tabelle.
29	Legge i dati da tabelle esterne Amazon S3.
30	Legge le informazioni sulle partizioni di una tabella esterna Amazon S3.
33	Legge i dati da una tabella Postgres remota.

ID di tipo	Descrizione
36	Legge i dati da una tabella MySQL remota.
37	Legge i dati da un flusso Kinesis remoto.

Note per l'utilizzo

Idealmente `rows` dovrebbe essere relativamente vicino a `rows_pre_filter`. Una grande differenza tra `rows` e `rows_pre_filter` indica che il motore di esecuzione sta eseguendo la scansione delle righe che verranno eliminate, il che non è efficiente. La differenza tra `rows_pre_filter` e `rows_pre_user_filter` è il numero di righe fantasma nella scansione. Esegui un VACUUM per rimuovere le righe contrassegnate per l'eliminazione. La differenza tra `rows` e `rows_pre_user_filter` è il numero di righe a cui la query ha applicato un filtro. Se il filtro dell'utente elimina molte righe, rivedi la tua scelta della colonna di ordinamento o, se ciò è dovuto a una grande regione non ordinata, esegui un vacuum.

Query di esempio

Il seguente esempio mostra che `rows_pre_filter` è maggiore di `rows_pre_user_filter` perché la tabella ha eliminato righe sulle quali non è stato eseguito un vacuum (righe fantasma).

```
SELECT query, slice, segment,step,rows, rows_pre_filter, rows_pre_user_filter
from stl_scan where query = pg_last_query_id();

 query |  slice | segment | step | rows  | rows_pre_filter | rows_pre_user_filter
-------+--------+---------+------+-------+-----------------+----------------------
 42915 |      0 |       0 |    0 | 43159 |           86318 |                43159
 42915 |      0 |       1 |    0 |     1 |               0 |                    0
 42915 |      1 |       0 |    0 | 43091 |           86182 |                43091
 42915 |      1 |       1 |    0 |     1 |               0 |                    0
 42915 |      2 |       0 |    0 | 42778 |           85556 |                42778
 42915 |      2 |       1 |    0 |     1 |               0 |                    0
 42915 |      3 |       0 |    0 | 43428 |           86856 |                43428
 42915 |      3 |       1 |    0 |     1 |               0 |                    0
 42915 |  10000 |       2 |    0 |     4 |               0 |                    0
(9 rows)
```

STL_SCHEMA_QUOTA_VIOLATIONS

Registra occorrenza, timestamp, XID e altre informazioni utili quando viene superata una quota dello schema.

STL_SCHEMA_QUOTA_VIOLATIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_SCHEMA_QUOTA_VIOLATIONS. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
ownerid	integer	ID del proprietario dello schema.
xid	bigint	L'ID di transazione associato all'istruzione.
pid	integer	L'ID di processo associato all'istruzione.
userid	integer	L'ID dell'utente che ha generato la voce.
schema_id	integer	Lo spazio dei nomi o l'ID dello schema.
schema_name	carattere (128)	Lo spazio dei nomi o il nome dello schema.
quota	integer	La quantità di spazio su disco (in MB) che lo schema può utilizzare.
disk_usage	integer	Lo spazio su disco (in MB) attualmente utilizzato dallo schema.
disk_usage_pct	double precision	Percentuale di spazio su disco attualmente utilizzata dallo schema al di fuori dalla quota configurata.

Nome colonna	Tipo di dati	Descrizione
timestamp	timestamp without time zone	L'ora in cui si è verificata la violazione.

Query di esempio

La query seguente mostra il risultato della violazione della quota:

```
SELECT userid, TRIM(SCHEMA_NAME) "schema_name", quota, disk_usage, disk_usage_pct,
 timestamp FROM
stl_schema_quota_violations WHERE SCHEMA_NAME = 'sales_schema' ORDER BY timestamp DESC;
```

Questa query restituisce il seguente output di esempio per lo schema specificato:

```
userid | schema_name  | quota | disk_usage | disk_usage_pct |timestamp
-------+--------------+-------+------------+----------------
+---------------------------
104    | sales_schema | 2048  | 2798       |  136.62        | 2020-04-20
 20:09:25.494723
(1 row)
```

STL_SESSIONS

Restituisce informazioni sulla cronologia della sessione utente.

STL_SESSIONS si differenzia da STV_SESSIONS nel fatto che STL_SESSIONS contiene la cronologia delle sessioni, mentre STV_SESSIONS contiene le sessioni attive correnti.

STL_SESSIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_SESSION_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere

più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
starttime	timestamp	Ora in UTC in cui è stata avviata la sessione.
endtime	timestamp	Ora in UTC in cui è stata terminata la sessione.
elaborazione	integer	ID di processo per la sessione.
user_name	character(50)	Nome utente associato alla sessione.
db_name	character(50)	Nome del database associato alla sessione.
timeout_sec	int	Il tempo massimo in secondi in cui una sessione rimane inattiva o inattiva prima del timeout. 0 indica che non è impostato alcun timeout.
timed_out	int	Un valore che indica se una sessione è scaduta: 1 se è scaduta, 0 in caso contrario.

Query di esempio

Per visualizzare la cronologia delle sessioni per il database TICKIT, digita la seguente query:

```
select starttime, process, user_name, timeout_sec, timed_out
from stl_sessions
where db_name='tickit' order by starttime;
```

Questa query restituisce il seguente output di esempio:

```
    starttime              | process |  user_name             | timeout_sec | timed_out
---------------------------+---------+------------------------+-------------
+-------------
2008-09-15 09:54:06.746705 |   32358 | dwuser                 | 120         | 1
2008-09-15 09:56:34.30275  |   32744 | dwuser                 | 60          | 1
2008-09-15 11:20:34.694837 |   14906 | dwuser                 | 0           | 0
2008-09-15 11:22:16.749818 |   15148 | dwuser                 | 0           | 0
2008-09-15 14:32:44.66112  |   14031 | dwuser                 | 0           | 0
2008-09-15 14:56:30.22161  |   18380 | dwuser                 | 0           | 0
2008-09-15 15:28:32.509354 |   24344 | dwuser                 | 0           | 0
2008-09-15 16:01:00.557326 |   30153 | dwuser                 | 120         | 1
2008-09-15 17:28:21.419858 |   12805 | dwuser                 | 0           | 0
2008-09-15 20:58:37.601937 |   14951 | dwuser                 | 60          | 1
2008-09-16 11:12:30.960564 |   27437 | dwuser                 | 60          | 1
2008-09-16 14:11:37.639092 |   23790 | dwuser                 | 3600        | 1
2008-09-16 15:13:46.02195  |    1355 | dwuser                 | 120         | 1
2008-09-16 15:22:36.515106 |    2878 | dwuser                 | 120         | 1
2008-09-16 15:44:39.194579 |    6470 | dwuser                 | 120         | 1
2008-09-16 16:50:27.02138  |   17254 | dwuser                 | 120         | 1
2008-09-17 12:05:02.157208 |    8439 | dwuser                 | 3600        | 0
(17 rows)
```

STL_SORT

Mostra le fasi di esecuzione di ordinamento per le query, come le fasi che utilizzano l'elaborazione ORDER BY.

STL_SORT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_SORT contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
byte	bigint	Dimensione, in byte, di tutte le righe di output della fase.
tbl	integer	ID tabella.
is_diskbased	character(1)	Se true (t), la query è stata eseguita come un'operazione basata su disco. Se false (f), la query è stata eseguita in memoria.

Nome colonna	Tipo di dati	Descrizione
workmem	bigint	Numero totale di byte nella memoria di lavoro che sono stati assegnati alla fase.
checksum	bigint	Queste informazioni sono solo per uso interno.

Query di esempio

L'esempio seguente restituisce i risultati di ordinamento per la sezione 0 e il segmento 1.

```
select query, bytes, tbl, is_diskbased, workmem
from stl_sort
where slice=0 and segment=1;
```

```
 query |  bytes  | tbl | is_diskbased |  workmem
-------+---------+-----+--------------+-----------
   567 | 3126968 | 241 | f            | 383385600
   604 |    5292 | 242 | f            | 383385600
   675 |  104776 | 251 | f            | 383385600
   525 | 3126968 | 251 | f            | 383385600
   585 |    5068 | 241 | f            | 383385600
   630 |  204808 | 266 | f            | 383385600
   704 |       0 | 242 | f            |         0
   669 | 4606416 | 241 | f            | 383385600
   696 |  104776 | 241 | f            | 383385600
   651 | 4606416 | 254 | f            | 383385600
   632 |       0 | 256 | f            |         0
   599 |     396 | 241 | f            | 383385600
 86397 |       0 | 242 | f            |         0
   621 |    5292 | 241 | f            | 383385600
 86325 |       0 | 242 | f            |         0
   572 |    5068 | 242 | f            | 383385600
   645 |  204808 | 241 | f            | 383385600
   590 |     396 | 242 | f            | 383385600
(18 rows)
```

STL_SSHCLIENT_ERROR

Registra tutti gli errori visti dal client SSH.

STL_SSHCLIENT_ERROR è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
recordtime	timestamp	Ora in cui l'errore è stato registrato.
pid	integer	Processo che ha registrato l'errore.
ssh_username	character (1024)	Il nome utente SSH.
endpoint	character (1024)	L'endpoint SSH.
command	character (4096)	Il comando SSH completo.
error	character (1024)	Messaggio di errore.

STL_STREAM_SEGS

Elenca la relazione tra flussi e segmenti simultanei.

I flussi in questo contesto sono flussi di Amazon Redshift. Questa vista di sistema non riguarda Importazione di dati in streaming.

STL_STREAM_SEGS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_STREAM_SEGS contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
stream	integer	Il set di segmenti simultanei di una query.
segment	integer	Numero identificativo del segmento di query.

Query di esempio

Per visualizzare la relazione tra flussi e segmenti simultanei per la query più recente, digita la seguente query:

```
select *
from stl_stream_segs
where query = pg_last_query_id();

 query | stream | segment
-------+--------+---------
    10 |      1 |       2
```

```
    10 |      0 |       0
    10 |      2 |       4
    10 |      1 |       3
    10 |      0 |       1
(5 rows)
```

STL_TR_CONFLICT

Mostra le informazioni per identificare e risolvere i conflitti di transazione con le tabelle di database.

Si verifica un conflitto di transazione quando due o più utenti stanno interrogando e modificando le righe di dati dalle tabelle così da impedire che le loro transazioni vengano serializzate. La transazione che esegue un'istruzione che interromperebbe la serializzabilità è interrotta e annullata. Ogni volta che si verifica un conflitto di transazione, Amazon Redshift scrive una riga di dati nella tabella di sistema STL_TR_CONFLICT contenente dettagli sulla transazione annullata. Per ulteriori informazioni, consulta Isolamento serializzabile.

STL_TR_CONFLICT è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_TRANSACTION_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
xact_id	bigint	ID di transazione per la transazione annullata.
process_id	bigint	Processo associato con la transazione che è stata annullata.
xact_start_ts	timestamp	Timestamp (UTC) in cui è iniziata la transazione.
abort_time	timestamp	Timestamp (UTC) in cui è stata arrestata la transazione.
table_id	bigint	ID di tabella per la tabella in cui si è verificato il conflitto.

Query di esempio

Per restituire informazioni sui conflitti che hanno coinvolto una tabella specifica, esegui una query che specifichi l'ID di tabella:

```
select * from stl_tr_conflict where table_id=100234
order by xact_start_ts;

xact_id|process_|      xact_start_ts       |        abort_time        |table_
       |id      |                          |                          |id
-------+--------+--------------------------+--------------------------+------
  1876 |  8551  |2010-03-30 09:19:15.852326|2010-03-30 09:20:17.582499|100234
  1928 | 15034  |2010-03-30 13:20:00.636045|2010-03-30 13:20:47.766817|100234
  1991 | 23753  |2010-04-01 13:05:01.220059|2010-04-01 13:06:06.94098 |100234
  2002 | 23679  |2010-04-01 13:17:05.173473|2010-04-01 13:18:27.898655|100234
(4 rows)
```

È possibile ottenere l'ID di tabella dalla sezione DETAIL del messaggio di errore per le violazioni di serializzabilità (errore 1023).

STL_UNDONE

Visualizza informazioni sulle transazioni annullate.

STL_UNDONE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_TRANSACTION_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
xact_id	bigint	ID per la transazione di annullamento.

Nome colonna	Tipo di dati	Descrizione
xact_id_undone	bigint	ID per la transazione annullata.
undo_start_ts	timestamp	Ora di inizio per la transazione di annullamento.
undo_end_ts	timestamp	Ora di fine per la transazione di annullamento.
table_id	bigint	ID per la tabella interessata dalla transazione di annullamento.

Query di esempio

Per visualizzare un log conciso di tutte le transazioni di annullamento, digita il comando seguente:

```
select xact_id, xact_id_undone, table_id from stl_undone;
```

Questo comando restituisce il seguente output di esempio:

```
 xact_id | xact_id_undone | table_id
---------+----------------+----------
1344 |           1344 |   100192
1326 |           1326 |   100192
1551 |           1551 |   100192
(3 rows)
```

STL_UNIQUE

Analizza le fasi di esecuzione che si verificano quando una funzione DISTINCT è utilizzata nell'elenco SELECT o quando i duplicati sono rimossi in una query UNION o INTERSECT.

STL_UNIQUE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_UNIQUE contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.

Nome colonna	Tipo di dati	Descrizione
righe	bigint	Numero totale di righe elaborate.
type	character(6)	Il tipo di fase. I valori validi sono: • HASHED. Indica che la fase ha utilizzato un'aggreg azione non ordinata, raggruppata. • PLAIN. Indica che la fase ha utilizzato un'aggregazione non raggruppata, scalare. • SORTED. Indica che la fase ha utilizzato un'aggreg azione ordinata, raggruppata.
is_diskba sed	character(1)	Se true (t), la query è stata eseguita come un'operazione basata su disco. Se false (f), la query è stata eseguita in memoria.
slots	integer	Numero totale di bucket di hash.
workmem	bigint	Numero totale di byte nella memoria di lavoro che sono stati assegnati alla fase.
max_buffe rs_used	bigint	Numero massimo di buffer utilizzati nella tabella di hash prima di andare al disco.
resizes	integer	Queste informazioni sono solo per uso interno.
occupied	integer	Queste informazioni sono solo per uso interno.
flushable	integer	Queste informazioni sono solo per uso interno.
used_uniq ue_prefet ching	character(1)	Queste informazioni sono solo per uso interno.
bytes	biginit	Numero di byte di tutte le righe di output per la fase.

Query di esempio

Supponi di eseguire la seguente query:

```
select distinct eventname
from event order by 1;
```

Ipotizzando che l'ID per la query precedente sia 6313, l'esempio seguente mostra il numero di righe prodotte dall'unica fase per ogni sezione nei segmenti 0 e 1.

```
select query, slice, segment, step, datediff(msec, starttime, endtime) as msec,
 tasknum, rows
from stl_unique where query = 6313
order by query desc, slice, segment, step;
```

```
 query | slice | segment | step | msec | tasknum | rows
-------+-------+---------+------+------+---------+------
  6313 |     0 |       0 |    2 |    0 |      22 |  550
  6313 |     0 |       1 |    1 |  256 |      20 |  145
  6313 |     1 |       0 |    2 |    1 |      23 |  540
  6313 |     1 |       1 |    1 |   42 |      21 |  127
  6313 |     2 |       0 |    2 |    1 |      22 |  540
  6313 |     2 |       1 |    1 |  255 |      20 |  158
  6313 |     3 |       0 |    2 |    1 |      23 |  542
  6313 |     3 |       1 |    1 |   38 |      21 |  146
(8 rows)
```

STL_UNLOAD_LOG

Registra i dettagli per un'operazione di scaricamento.

STL_UNLOAD_LOG registra una riga per ogni file creato da un'istruzione UNLOAD. Ad esempio, se un UNLOAD crea 12 file, STL_UNLOAD_LOG conterrà 12 righe corrispondenti.

STL_UNLOAD_LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_UNLOAD_LOG contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_UNLOAD_HISTORY e SYS_UNLOAD_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	L'ID di query.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
pid	integer	ID di processo associato all'istruzione di query.
path	character(1280)	Il percorso completo di un oggetto Amazon S3 per il file.
start_time	timestamp	Ora di inizio per la transazione.
end_time	timestamp	Ora di fine per la transazione.
line_count	bigint	Numero di righe scaricate nel file.
transfer_ size	bigint	Numero di byte trasferiti.
file_format	character(10)	Formato del file scaricato.

Query di esempio

Per ottenere un elenco dei file scritti in Amazon S3 da un comando UNLOAD, puoi chiamare un'operazione di elenco Amazon S3 dopo il completamento di UNLOAD. È inoltre possibile eseguire una query su STL_UNLOAD_LOG.

La seguente query restituisce il percorso per i file creati da un UNLOAD per l'ultima query completata:

```
select query, substring(path,0,40) as path
from stl_unload_log
where query = pg_last_query_id()
order by path;
```

Questo comando restituisce il seguente output di esempio:

```
 query |             path
-------+--------------------------------------
  2320 | s3://my-bucket/venue0000_part_00
  2320 | s3://my-bucket/venue0001_part_00
  2320 | s3://my-bucket/venue0002_part_00
  2320 | s3://my-bucket/venue0003_part_00
(4 rows)
```

STL_USAGE_CONTROL

La vista STL_USAGE_CONTROL contiene informazioni che vengono registrate quando viene raggiunto un limite di utilizzo. Per ulteriori informazioni sui limiti di utilizzo, consulta Gestione dei limiti di utilizzo nella Guida alla gestione di Amazon Redshift.

STL_USAGE_CONTROL è visibile solo per gli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
eventtime	timestamp	L'ora (UTC) in cui la query ha superato un limite di utilizzo.
query	integer	L'identificativo della query. È possibile utilizzare questo ID per unire varie altre tabelle e visualizzazioni di sistema.

Nome colonna	Tipo di dati	Descrizione
xid	bigint	L'identificativo della transazione.
pid	integer	Identificativo di processo associato alla query.
usage_limit_id	character(40)	Un identificativo univoco universale (UUID) generato da Amazon Redshift, ad esempio `25d9297e-3e7b-41c8-9f4d-c4b6eb731c09` .
feature_type	character(30)	Caratteristica il cui limite di utilizzo è stato superato. I valori possibili sono `CONCURRENCY_SCALING` e `SPECTRUM`.

Query di esempio

Nell'esempio SQL seguente vengono restituite alcune delle informazioni registrate quando viene raggiunto un limite di utilizzo.

```
select query, pid, eventtime, feature_type
from stl_usage_control
order by eventtime desc
limit 5;
```

STL_USERLOG

Registra i dettagli per le seguenti modifiche a un utente di database:

- Create user (Crea utente)
- Rimozione dell'utente
- Modifica di un utente (assegnazione di un nuovo nome)
- Modifica di un utente (modifica delle proprietà)

STL_USERLOG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_USERLOG. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente interessato dalla modifica.
username	character(50)	Nome utente dell'utente interessato dalla modifica.
oldusername	character(50)	Per un'operazione di assegnazione di un nuovo nome, il nome utente originale. Per ogni altra operazione, questo campo è vuoto.
action	character(10)	Operazione che si è verificata. Valori validi: • Alter • Crea • Drop (E-mail eliminata) • Assegnazione di un nuovo nome
usecreatedb	integer	Se true (1), indica che l'utente ha creato dei privilegi di database.
usesuper	integer	Se true (1), indica che l'utente è un utente con privilegi avanzati.
usecatupd	integer	Se true (1), indica che l'utente può aggiornare i cataloghi di sistema.
valuntil	timestamp	Data di scadenza della password.
pid	integer	ID processo.
xid	bigint	ID transazione.
recordtime	timestamp	Ora in UTC in cui è stata avviata la query.

Query di esempio

Il seguente esempio esegue quattro operazioni da parte dell'utente, poi interroga la vista STL_USERLOG.

```
create user userlog1 password 'Userlog1';
alter user userlog1 createdb createuser;
alter user userlog1 rename  to userlog2;
drop user userlog2;

select userid, username, oldusername, action, usecreatedb, usesuper from stl_userlog
 order by recordtime desc;
```

```
 userid |  username | oldusername |  action | usecreatedb | usesuper
--------+-----------+-------------+---------+-------------+----------
    108 | userlog2  |             | drop    |           1 |   1
    108 | userlog2  | userlog1    | rename  |           1 |   1
    108 | userlog1  |             | alter   |           1 |   1
    108 | userlog1  |             | create  |           0 |   0
 (4 rows)
```

STL_UTILITYTEXT

Acquisisce il testo dei comandi SQL non-SELECT eseguiti sul database.

Eseguire una query sulla vista STL_UTILITYTEXT per acquisire il seguente sottoinsieme di istruzioni SQL che sono state eseguite nel sistema:

- ABORT, BEGIN, COMMIT, END, ROLLBACK
- ANALYZE
- CALL
- CANCEL
- COMMENT
- CREATE, ALTER, DROP DATABASE
- CREATE, ALTER, DROP USER

- EXPLAIN
- GRANT, REVOKE
- LOCK
- RESET
- SET
- MOSTRA
- TRUNCATE

Consulta anche STL_DDLTEXT, STL_QUERYTEXT e SVL_STATEMENTTEXT.

Utilizza le colonne STARTTIME ed ENDTIME per scoprire quali istruzioni sono state registrate in un dato periodo. Blocchi lunghi di testo SQL sono suddivisi in righe di 200 caratteri; la colonna SEQUENCE identifica i frammenti di testo che appartengono a ogni singola istruzione.

STL_UTILITYTEXT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
xid	bigint	ID transazione.
pid	integer	ID di processo associato all'istruzione di query.
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etiche tta definita con un comando SET QUERY GROUP. Se la query non è basata su file o il parametro QUERY_GROUP non è impostato, questo campo è vuoto.

Nome colonna	Tipo di dati	Descrizione
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358** .
sequenza	integer	Quando una singola istruzione contiene più di 200 caratteri , vengono registrate delle righe aggiuntive per tale istruzion e. La sequenza 0 è la prima riga, 1 la seconda e così via.
text	character(200)	Testo SQL, in incrementi da 200 caratteri. Questo campo potrebbe contenere caratteri speciali come barra rovesciat a (\\) e nuova riga (\n).

Query di esempio

La seguente query restituisce il testo per i comandi "utility" eseguiti il 26 gennaio 2012. In questo caso, sono stati eseguiti alcuni comandi SET e un comando SHOW ALL:

```
select starttime, sequence, rtrim(text)
from stl_utilitytext
where starttime like '2012-01-26%'
order by starttime, sequence;

starttime          | sequence |             rtrim
---------------------------+-----+---------------------------------
2012-01-26 13:05:52.529235 |   0 | show all;
2012-01-26 13:20:31.660255 |   0 | SET query_group to ''
2012-01-26 13:20:54.956131 |   0 | SET query_group to 'soldunsold.sql'
...
```

Ricostruzione dell'SDL archiviato

Per ricostruire l'SQL archiviato nella colonna `text` di STL_UTILITYTEXT, eseguire un'istruzione SELECT per creare un SQL da 1 o più parti nella colonna `text`. Prima di eseguire l'SQL ricostruito, sostituire un carattere speciale qualsiasi (\n) con una nuova riga. Il risultato della dichiarazione SELECT seguente è dato da una riga di SQL ricostruiti nel campo `query_statement`.

```
SELECT LISTAGG(CASE WHEN LEN(RTRIM(text)) = 0 THEN text ELSE RTRIM(text) END) WITHIN
 GROUP (ORDER BY sequence) as query_statement
FROM stl_utilitytext GROUP BY xid order by xid;
```

Per esempio, la query seguente imposta query_group su una sequenza di zeri. La query stessa è più lunga di 200 caratteri ed è archiviata in parti in STL_UTILITYTEXT.

```
set query_group to
 '0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
             000000';
```

In questo esempio, la query è archiviata in 2 parti (righe) nella colonna `text` di STL_UTILITYTEXT.

```
select query, sequence, text
from stl_utilitytext where query=pg_last_query_id() order by query desc, sequence limit
 10;
```

```
         starttime          | sequence |
                                                text

----------------------------+----------
+-------------------------------------------------------------------------------------------------
 2019-07-23 22:55:34.926198 |        0 | set query_group to
 '0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\n00000000000
 2019-07-23 22:55:34.926198 |        1 |                   000000';
```

Per ricostruire l'SQL archiviato in STL_UTILITYTEXT, eseguire l'SQL seguente.

```
select LISTAGG(CASE WHEN LEN(RTRIM(text)) = 0 THEN text ELSE RTRIM(text) END, '')
 within group (order by sequence) AS query_statement
```

```
from stl_utilitytext where query=pg_last_query_id();
```

Per utilizzare l'SQL ricostruito derivante nel client, sostituire qualsiasi carattere speciale (\n) con una nuova riga.

```
                                                query_statement


-------------------------------------------------------------------------------------------------------
 set query_group to
 '0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\n00000000000
                  000000';
```

STL_VACUUM

Visualizza statistiche di riga e di blocco per le tabelle sottoposte a vacuum.

La vista mostra informazioni relative al momento in cui ogni operazione di vacuum è stata avviata o terminata e dimostra i benefici nell'esecuzione dell'operazione. Per informazioni sui requisiti per l'esecuzione di questo comando, consultare la descrizione di comando VACUUM.

STL_VACUUM è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_VACUUM_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato la voce.
xid	bigint	ID di transazione per l'istruzione VACUUM. Puoi collegare questa tabella alla vista STL_QUERY per visualizzare le singole istruzioni SQL eseguite per una

Nome colonna	Tipo di dati	Descrizione
		determinata transazione VACUUM. Se sottoponi a vacuum l'intero database, ogni tabella è sottoposta a vacuum in una transazione separata.
table_id	integer	L'ID della tabella.

Nome colonna	Tipo di dati	Descrizione
status	character(30)	Lo stato dell'operazione VACUUM per ogni tabella. I valori possibili sono i seguenti: • **Started** • **Started Delete Only** • **Started Delete Only (Sorted >= nn%)** Solo la fase di eliminazione è stata avviata per un VACUUM FULL. La fase di ordinamento è stata ignorata perché la tabella era stata già ordinata alla soglia di ordinamento o al di sopra di essa. • **Started Sort Only** • **Started Ranged Partition** • **Started Reindex** • **Finished** Ora in cui l'operazione è stata completata per la tabella. Per scoprire quanto tempo ha impiegato un'operazione di vacuum su una particolare tabella, sottrai l'ora di inizio dall'ora di fine per un ID di transazione e un ID di tabella specifici. • **Skipped** La tabella è stata ignorata perché era stata completam ente ordinata e non erano state contrassegnate righe per l'eliminazione. • **Skipped (delete only)** La tabella è stata ignorata perché è stato specificato DELETE ONLY e non sono state contrassegnate righe per l'eliminazione. • **Skipped (sort only)**

Nome colonna	Tipo di dati	Descrizione
		La tabella è stata ignorata perché è stato specificato SORT ONLY e la tabella era già stata completamente ordinata. • **`Skipped (sort only, sorted>=xx%)`** La tabella è stata ignorata perché è stato specificato SORT ONLY e la tabella era già stata ordinata alla soglia di ordinamento o al di sopra di essa. • **`Skipped (0 rows)`** La tabella è stata ignorata perché vuota. • **`VacuumBG`** Un'operazione vacuum automatica è stata eseguita in background. Questo stato viene anteposto agli altri stati quando vengono eseguiti automaticamente. Ad esempio, un'operazione vacuum di sola eliminazione eseguita automaticamente avrebbe una riga iniziale con lo stato `[VacuumBG] Started Delete Only` . Per ulteriori informazioni sull'impostazione della soglia di ordinamento di VACUUM, consultare VACUUM.
righe	bigint	L'effettivo numero di righe nella tabella più le righe eliminate ancora archiviate su disco (in attesa di essere sottoposte a vacuum). Questa colonna mostra il conteggio prima dell'avvio del vacuum per le righe con uno stato **`Started`** e il conteggio dopo il vacuum per le righe con uno stato **`Finished`**.

Nome colonna	Tipo di dati	Descrizione
sortedrows	integer	Il numero di righe che sono state ordinate nella tabella. Questa colonna mostra il conteggio prima dell'avvio del vacuum per le righe con uno stato **`Started`** nella colonna di stato e il conteggio dopo il vacuum per le righe con uno stato **`Finished`** nella colonna di stato.
blocks	integer	Il numero totale di blocchi di dati utilizzati per archiviar e i dati della tabella prima dell'operazione di vacuum (righe con uno stato **`Started`**) e dopo di essa (colonna **`Finished`**). Ogni blocco di dati utilizza 1 MB.
max_merge_partitio ns	integer	Questa colonna viene utilizzata per l'analisi delle prestazioni e rappresenta il numero massimo di partizion i che il vacuum può elaborare per la tabella per ogni iterazione di fase di unione. (Il vacuum ordina la regione non ordinata in una o più partizioni ordinate. A seconda del numero di colonne nella tabella e della configura zione attuale di Amazon Redshift, la fase di unione può elaborare un numero massimo di partizioni in una singola interazione di unione. La fase di unione continuerà a funzionare se il numero di partizioni ordinate supera il numero massimo di partizioni di unione, ma saranno necessarie altre iterazioni di unione).
eventtime	timestamp	Il momento in cui l'operazione di vacuum è stata avviata o terminata.
reclaimable_rows	bigint	Il numero di righe recuperabili per il cutoff_xid corrente. Questa colonna mostra il numero stimato di righe recuperabili di Redshift prima dell'inizio dell'operazione vacuum per le righe con uno stato **`Started`** e il numero effettivo di righe recuperabili rimanenti dopo l'operazione vacuum per le righe con uno stato **`Finished`**.

Nome colonna	Tipo di dati	Descrizione
reclaimable_space_mb	bigint	Spazio recuperabile in MB per il cutoff_xid corrente. Questa colonna mostra la quantità di spazio recuperab ile stimata di Redshift prima dell'inizio dell'operazione vacuum per le righe con uno stato **`Started`** e la quantità effettiva di spazio recuperabile rimanente dopo l'operazi one vacuum per le righe con uno stato **`Finished`**.
cutoff_xid	bigint	ID di transazione limite dell'operazione VACUUM. Eventuali transazioni successive al limite non sono incluse nell'operazione VACUUM.
is_recluster	integer	Se 1 (true), l'operazione VACUUM ha eseguito l'algoritmo recluster; se 0 (false), non c'è stata alcuna esecuzione.

Query di esempio

La seguente query dichiara le statistiche di vacuum per la tabella 108313. La tabella è stata sottoposta a vacuum in seguito a una serie di inserimenti ed eliminazioni.

```
select xid, table_id, status, rows, sortedrows, blocks, eventtime,
                reclaimable_rows, reclaimable_space_mb
from stl_vacuum where table_id=108313 order by eventtime;

 xid   | table_id | status                  | rows | sortedrows | blocks |  eventtime
         | reclaimable_rows | reclaimable_space_mb
-------+----------+-------------------------+------+------------+--------
+----------------------+------------------+----------------------
 14294 |   108313 | Started                 | 1950 |        408 |     28 |  2016-05-19
 17:36:01 |              984 |                   17
 14294 |   108313 | Finished                |  966 |        966 |     11 |  2016-05-19
 18:26:13 |                0 |                    0
 15126 |   108313 | Skipped(sorted>=95%)    |  966 |        966 |     11 |  2016-05-19
 18:26:38 |                0 |                    0
```

All'inizio dell'operazione VACUUM, la tabella conteneva 1.950 righe archiviate in 28 blocchi da 1 MB. Amazon Redshift ha stimato di poter recuperare 984 righe, o 17 blocchi di spazio su disco, con un'operazione vacuum.

Nella riga dello stato Finished, la colonna ROWS mostra un valore di 966 e il valore della colonna BLOCKS è 11, rispetto a 28. L'operazione vacuum ha recuperato la quantità stimata di spazio su disco, senza righe o spazio recuperabili restanti dopo il completamento dell'operazione vacuum.

Nella fase di ordinamento (transazione 15126), il vacuum è riuscito a ignorare la tabella perché le righe sono state inserite nell'ordine della chiave di ordinamento.

Il seguente esempio mostra le statistiche per un vacuum SORT ONLY sulla tabella SALES (tabella 110116 in questo esempio) dopo una lunga operazione INSERT:

```
vacuum sort only sales;

select xid, table_id, status, rows, sortedrows, blocks, eventtime
from stl_vacuum order by xid, table_id, eventtime;

xid |table_id|     status      | rows  |sortedrows|blocks|     eventtime
----+--------+-----------------+-------+----------+------+--------------------
...
2925| 110116 |Started Sort Only|1379648|   172456 |  132 | 2011-02-24 16:25:21...
2925| 110116 |Finished         |1379648|  1379648 |  132 | 2011-02-24 16:26:28...
```

STL_WINDOW

Analizza le fasi di query che eseguono funzioni finestra.

STL_WINDOW è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> STL_WINDOW contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
sezione	integer	Numero che identifica la sezione in cui è stata eseguita la query.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
starttime	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'orario in UTC in cui è terminata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .
tasknum	integer	Il numero del processo di attività di query assegnato per eseguire la fase.
righe	bigint	Numero totale di righe elaborate.
is_diskbased	character(1)	Se true (t), la query è stata eseguita come un'operazione basata su disco. Se false (f), la query è stata eseguita in memoria.
workmem	bigint	Numero totale di byte nella memoria di lavoro che sono stati assegnati alla fase.

Query di esempio

L'esempio seguente restituisce i risultati della funzione finestra per la sezione 0 e il segmento 3.

```
select query, tasknum, rows, is_diskbased, workmem
from stl_window
where slice=0 and segment=3;
```

```
 query | tasknum | rows | is_diskbased | workmem
-------+---------+------+--------------+----------
 86326 |      36 | 1857 | f            | 95256616
   705 |      15 | 1857 | f            | 95256616
 86399 |      27 | 1857 | f            | 95256616
   649 |      10 |    0 | f            | 95256616
(4 rows)
```

STL_WLM_ERROR

Registra tutti gli errori relativi a WLM mentre si verificano.

STL_WLM_ERROR è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
recordtime	timestamp	Ora in cui l'errore si è verificato.
pid	integer	ID per il processo che ha generato l'errore.
error_string	character(256)	Descrizione dell'errore.

STL_WLM_RULE_ACTION

Registra i dettagli sulle operazioni derivanti dalle regole di monitoraggio di query WLM associate alle code definite dall'utente. Per ulteriori informazioni, consulta Regole di monitoraggio delle query WLM.

STL_WLM_RULE_ACTION è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	Utente che ha eseguito la query.
query	integer	ID query.
service_class	integer	ID per la classe di servizio. Le code di query sono definite nella configurazione WLM. Classi di servizio maggiori di 5 sono code definite dall'utente.
rule	character(256)	Nome di una regola di monitoraggio delle query.
action	character(256)	Operazione risultante. I valori possibili sono i seguenti: • log • hop(reassign) • hop(restart) • abort • change_query_priority • nessuno Un valore none indica che i predicati della regola sono soddisfatti ma l'operazione è stata sostituita da un'altra regola con un'operazione con gravità più alta.
recordtime	timestamp	Ora in cui l'azione è stata registrata in UTC.

Nome colonna	Tipo di dati	Descrizione
action_value	character(256)	Se `action` è `change_query_priority` , i valori possibili sono `highest`, `high`, `normal`, `low` e `lowest`. Se `action` è `log`, hop o `abort`, il valore è vuoto.
service_class_name	character(64)	Il nome della classe dei servizi.

Query di esempio

L'esempio seguente trova le query interrotte da una regola di monitoraggio delle query.

```
Select query, rule
from stl_wlm_rule_action
where action = 'abort'
order by query;
```

STL_WLM_QUERY

Contiene un record per ogni tentativo di esecuzione di una query in una classe di servizio gestita da WLM.

STL_WINDOW è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.

Nome colonna	Tipo di dati	Descrizione
xid	integer	ID di transazione della query o della subquery.
task	integer	ID utilizzato per tenere traccia di una query via il gestore del carico di lavoro. Può essere associato a più ID di query. Se una query viene riavviata, alla stessa viene assegnato un nuovo ID di query ma non un nuovo ID di attività.
query	integer	ID query. Se una query viene riavviata, alla stessa viene assegnato un nuovo ID di query ma non un nuovo ID di attività.
service_class	integer	ID per la classe di servizio. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.
slot_count	integer	Numero di slot di query WLM utilizzati da una query in base al livello di simultaneità impostato per la coda. Il valore predefinito è 1. Per ulteriori informazi oni, consulta wlm_query_slot_count.
service_class_start_time	timestamp	Ora in cui la query è stata assegnata alla classe di servizio. Questo orario è nel fuso orario UTC.
queue_start_time	timestamp	Ora in cui la query è stata inserita nella coda per la classe di servizio. Questo orario è nel fuso orario UTC.
queue_end_time	timestamp	Ora in cui la query ha lasciato la coda per la classe di servizio. Questo orario è nel fuso orario UTC.
total_queue_time	bigint	Numero totale di microsecondi che la query ha trascorso nella coda
exec_start_time	timestamp	Ora in cui la query ha iniziato l'esecuzione nella classe di servizio. Questo orario è nel fuso orario UTC.

Nome colonna	Tipo di dati	Descrizione
exec_end_time	timestamp	Ora in cui la query ha completato l'esecuzione nella classe di servizio. Questo orario è nel fuso orario UTC.
total_exec_time	bigint	Numero di microsecondi che la query ha impiegato per l'esecuzione.
service_class_end_time	timestamp	Ora in cui la query ha lasciato la classe di servizio. Questo orario è nel fuso orario UTC.
final_state	character(16)	Riservato per il sistema.
est_peak_mem	bigint	Riservato per il sistema.
query_priority	char(20)	La priorità della query. I valori possibili sono `n/a`, `lowest`, `low`, `normal`, `high` e `highest`, dove `n/a` indica che la priorità della query non è supportata.
service_class_name	character(64)	Il nome della classe di servizio. Per ulteriori informazioni sulle classi di servizio, consulta Tabelle e viste di sistema di WLM.

Query di esempio

Visualizzazione del tempo medio delle query in coda e in esecuzione

Le seguenti query mostrano la configurazione corrente per le classi di servizio maggiori di 4. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.

La seguente query restituisce il tempo medio (in microsecondi) che ogni query ha trascorso nelle code e in esecuzione per ogni classe di servizio.

```
select service_class as svc_class, count(*),
avg(datediff(microseconds, queue_start_time, queue_end_time)) as avg_queue_time,
avg(datediff(microseconds, exec_start_time, exec_end_time )) as avg_exec_time
from stl_wlm_query
where service_class > 4
group by service_class
```

```
order by service_class;
```

Questa query restituisce il seguente output di esempio:

```
 svc_class | count | avg_queue_time | avg_exec_time
-----------+-------+----------------+---------------
         5 | 20103 |              0 |         80415
         5 |  3421 |          34015 |        234015
         6 |    42 |              0 |        944266
         7 |   196 |           6439 |       1364399
(4 rows)
```

Visualizzazione del tempo massimo delle query in coda e in esecuzione

La seguente query restituisce la quantità massima di tempo (in microsecondi) che una query ha trascorso in ogni coda di query e in esecuzione per ogni classe di servizio.

```
select service_class as svc_class, count(*),
max(datediff(microseconds, queue_start_time, queue_end_time)) as max_queue_time,
max(datediff(microseconds, exec_start_time, exec_end_time )) as max_exec_time
from stl_wlm_query
where svc_class > 5
group by service_class
order by service_class;
```

```
 svc_class | count | max_queue_time | max_exec_time
-----------+-------+----------------+---------------
         6 |    42 |              0 |       3775896
         7 |   197 |          37947 |      16379473
(4 rows)
```

Tabelle STV per dati di snapshot

Le tabelle STV sono tabelle di sistema virtuali che contengono snapshot dei dati di sistema correnti.

Argomenti

- STV_ACTIVE_CURSORS
- STV_BLOCKLIST
- STV_CURSOR_CONFIGURATION
- STV_DB_ISOLATION_LEVEL

- STV_EXEC_STATE
- STV_INFLIGHT
- STV_LOAD_STATE
- STV_LOCKS
- STV_ML_MODEL_INFO
- STV_MV_DEPS
- STV_MV_INFO
- STV_NODE_STORAGE_CAPACITY
- STV_PARTITIONS
- STV_QUERY_METRICS
- STV_RECENTS
- STV_SESSIONS
- STV_SLICES
- STV_STARTUP_RECOVERY_STATE
- STV_TBL_PERM
- STV_TBL_TRANS
- STV_WLM_CLASSIFICATION_CONFIG
- STV_WLM_QMR_CONFIG
- STV_WLM_QUERY_QUEUE_STATE
- STV_WLM_QUERY_STATE
- STV_WLM_QUERY_TASK_STATE
- STV_WLM_SERVICE_CLASS_CONFIG
- STV_WLM_SERVICE_CLASS_STATE
- STV_XRESTORE_ALTER_QUEUE_STATE

STV_ACTIVE_CURSORS

STV_ACTIVE_CURSORS visualizza i dettagli dei cursori correntemente aperti. Per ulteriori informazioni, consultare DECLARE.

STV_ACTIVE_CURSORS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità

dei dati nelle tabelle e nelle viste di sistema. Un utente può visualizzare soltanto i cursori che ha aperto. un utente con privilegi avanzati può visualizzare tutti i cursori.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
nome	characte (256)	Nome di cursore.
xid	bigint	Contesto della transazione.
pid	integer	Processo principale che esegue la query.
starttime	timestam	Ora in cui il cursore è stato dichiarato.
row_count	bigint	Numero di righe nel set di risultati del cursore.
byte_count	bigint	Numero di byte nel set di risultati del cursore.
fetched_r ows	bigint	Numero di righe correntemente recuperate dal set di risultati del cursore.

STV_BLOCKLIST

STV_BLOCKLIST contiene il numero di blocchi di disco da 1 MB utilizzati da ogni sezione, tabella o colonna in un database.

Per determinare il numero di blocchi di disco da 1 MB allocati per database, tabella, sezione o colonna, utilizza query di aggregazione con STV_BLOCKLIST, come illustrato negli esempi seguenti. È possibile anche utilizzare STV_PARTITIONS per visualizzare informazioni di riepilogo sull'utilizzo del disco.

STV_BLOCKLIST è visibile solo per agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
sezione	integer	Sezione del nodo.
col	integer	Indice in base zero della colonna. A ogni tabella che crei vengono aggiunte tre colonne nascoste: INSERT_XID, DELETE_XID e ROW_ID (OID). Una tabella con 3 colonne definite dall'utente contiene 6 colonne effettive e le colonne definite dall'utente sono numerate 0, 1 e 2. In questo esempio, le colonne INSERT_XID, DELETE_XID e ROW_ID sono numerate 3, 4 e 5 rispettivamente.
tbl	integer	ID di tabella della tabella di database.
blocknum	integer	ID del blocco di dati.
num_values	integer	Numero di valori contenuti nel blocco.
extended_limits	integer	Per uso interno.
minvalue	bigint	Valore di dati minimo del blocco. Memorizza i primi otto caratteri come valore integer a 64 bit per dati non numerici. Utilizzato per la scansione del disco.
maxvalue	bigint	Valore di dati massimo del blocco. Memorizza i primi otto caratteri come valore integer a 64 bit per dati non numerici. Utilizzato per la scansione del disco.
sb_pos	integer	Identificatore interno di Amazon Redshift per la posizione del super blocco sul disco.
pinned	integer	Indica se il blocco è aggiunto o meno nella memoria come parte del precaricamento. 0 = false; 1 = true. Il valore predefinito è false.
on_disk	integer	Indica se il blocco è archiviato automaticamente sul disco o meno. 0 = false; 1 = true. Il valore predefinito è false.

Nome colonna	Tipo di dati	Descrizione
modified	integer	Indica se il blocco è stato modificato o meno. 0 = false; 1 = true. Il valore predefinito è false.
hdr_modified	integer	Indica se l'intestazione del blocco è stata modificata o meno. 0 = false; 1 = true. Il valore predefinito è false.
unsorted	integer	Indica se il blocco è non ordinato o meno. 0 = false; 1 = true. Il valore predefinito è true.
tombstone	integer	Per uso interno.
preferred_diskno	integer	Numero di disco su cui il blocco deve trovarsi, salvo in caso di errore del disco. Dopo la correzione del disco, il blocco ritorna su questo disco.
temporary	integer	Indica se il blocco contiene o meno dati temporanei, ad esempio di una tabella temporanea o di risultati intermedi delle query. 0 = false; 1 = true. Il valore predefinito è false.
newblock	integer	Indica se un blocco è nuovo (true) o se non ne è mai stato eseguito il commit sul disco (false). 0 = false; 1 = true.
num_readers	integer	Numero di riferimenti su ogni blocco.
flags	integer	Flag interni di Amazon Redshift per l'intestazione del blocco.

Query di esempio

STV_BLOCKLIST contiene una riga per blocco di disco allocato, di conseguenza una query che seleziona tutte le righe può restituire un gran numero di righe. Ti consigliamo di utilizzare solo query di aggregazione con STV_BLOCKLIST.

La vista SVV_DISKUSAGE fornisce informazioni simili in un formato più semplice; tuttavia, l'esempio seguente illustra un utilizzo della tabella STV_BLOCKLIST.

Per determinare il numero di blocchi da 1 MB utilizzati da ogni colonna nella tabella VENUE, digita la seguente query:

```
select col, count(*)
from stv_blocklist, stv_tbl_perm
where stv_blocklist.tbl = stv_tbl_perm.id
and stv_blocklist.slice = stv_tbl_perm.slice
and stv_tbl_perm.name = 'venue'
group by col
order by col;
```

Questa query restituisce il numero di blocchi da 1 MB allocati a ogni colonna nella tabella VENUE, come mostrato nei seguenti esempi di dati:

```
 col | count
-----+-------
   0 |  4
   1 |  4
   2 |  4
   3 |  4
   4 |  4
   5 |  4
   7 |  4
   8 |  4
(8 rows)
```

La seguente query indica se i dati della tabella sono effettivamente distribuiti a tutte le sezioni:

```
select trim(name) as table, stv_blocklist.slice, stv_tbl_perm.rows
from stv_blocklist,stv_tbl_perm
where stv_blocklist.tbl=stv_tbl_perm.id
and stv_tbl_perm.slice=stv_blocklist.slice
and stv_blocklist.id > 10000 and name not like '%#m%'
and name not like 'systable%'
group by name, stv_blocklist.slice, stv_tbl_perm.rows
order by 3 desc;
```

Questa query produce il seguente output di esempio, che illustra la distribuzione omogenea dei dati della tabella con la maggior parte delle righe:

```
table   | slice | rows
----------+-------+-------
listing  |    13 | 10527
```

```
listing  |    14 | 10526
listing  |     8 | 10526
listing  |     9 | 10526
listing  |     7 | 10525
listing  |     4 | 10525
listing  |    17 | 10525
listing  |    11 | 10525
listing  |     5 | 10525
listing  |    18 | 10525
listing  |    12 | 10525
listing  |     3 | 10525
listing  |    10 | 10525
listing  |     2 | 10524
listing  |    15 | 10524
listing  |    16 | 10524
listing  |     6 | 10524
listing  |    19 | 10524
listing  |     1 | 10523
listing  |     0 | 10521
...
(180 rows)
```

La seguente query determina se è stato eseguito il commit sul disco di qualsiasi blocco rimosso definitivamente:

```
select slice, col, tbl, blocknum, newblock
from stv_blocklist
where  tombstone > 0;

slice | col |   tbl  | blocknum | newblock
-------+-----+--------+----------+----------
4     |  0  | 101285 |    0     |   1
4     |  2  | 101285 |    0     |   1
4     |  4  | 101285 |    1     |   1
5     |  2  | 101285 |    0     |   1
5     |  0  | 101285 |    0     |   1
5     |  1  | 101285 |    0     |   1
5     |  4  | 101285 |    1     |   1
...
(24 rows)
```

STV_CURSOR_CONFIGURATION

STV_CURSOR_CONFIGURATION visualizza i vincoli di configurazione del cursore. Per ulteriori informazioni, consultare Vincoli del cursore.

STV_CURSOR_CONFIGURATION è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
current_cursor_count	integer	Numero di cursori correntemente aperti.
max_diskspace_usable	integer	Quantità di spazio su disco disponibile per i cursori in megabyte. Questo vincolo è basato sulla dimensione del set di risultati del cursore massimo per il cluster.
current_diskspace_used	integer	Quantità di spazio su disco correntemente utilizzata dai cursori in megabyte.

STV_DB_ISOLATION_LEVEL

STV_DB_ISOLATION_LEVEL visualizza il livello di isolamento corrente per i database. Per ulteriori informazioni sui livelli di isolamento, consulta CREATE DATABASE.

STV_DB_ISOLATION_LEVEL è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
db_name	characte (128)	Nome del database.
isolation _level	characte (20)	Il livello di isolamento del database. I valori possibili sono `Serializa ble` e `Snapshot Isolation` .

STV_EXEC_STATE

Utilizza la tabella STV_EXEC_STATE per trovare informazioni sulle query e sulle fasi delle query in corso di esecuzione sui nodi di calcolo.

Questa informazioni sono in genere utilizzate solo per risolvere problemi tecnici. Le viste SVV_QUERY_STATE e SVL_QUERY_SUMMARY ottengono le relative informazioni da STV_EXEC_STATE.

STV_EXEC_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizzazioni di sistema.
sezione	integer	La sezione del nodo dove è stata completata la fase.

Nome colonna	Tipo di dati	Descrizione
segment	integer	Segmento delle query che è stata eseguita. Un segment di query è una serie di fasi.
step	integer	Fase del segmento di query che è stato eseguito. Una fase è la più piccola unità eseguita da una query.
starttime	timestamp	Ora che la fase è stata eseguita.
currenttime	timestamp	Ora corrente.
tasknum	integer	Processo di attività di query assegnato per completare la fase.
righe	bigint	Numero di righe elaborate.
byte	bigint	Numero di byte elaborati.
etichetta	char(256)	Etichetta di fase, che consiste in un nome di fase di query e, quando applicabile, in un ID di tabella e in un nome di tabella (per esempio, `scan tbl=100448 name =user`). Gli ID di tabella a tre cifre fanno in genere riferimento alle scansioni delle tabelle transitorie. Quando viene visualizzato `tbl=0`, fa in genere riferimento a una scansione di un valore costante.
is_diskbased	char(1)	Se questa fase della query è stata completata come operazione basata su disco: true (**`t`**) o false (**`f`**). Solo determinate fasi, come hash, sort e le fasi di aggregazione, possono accedere al disco. Molti tipi di fase sono sempre completati in memoria.
workmem	bigint	Numero di byte della memoria di lavoro assegnati alla fase.

Nome colonna	Tipo di dati	Descrizione
num_parts	integer	Numero di partizioni in cui una tabella di hash è divisa durante una fase di hash. Un numero positivo in questa colonna non implica che la fase di hash sia stata eseguita come operazione basata su disco. Verifica il valore nella colonna IS_DISKBASED per determinare se la fase di hash era basata sul disco.
is_rrscan	char(1)	Se true (**t**), indica che in questa fase è stata utilizzata la scansione a intervallo limitato. Il valore predefinito è false (**f**).
is_delayed_scan	char(1)	Se true (**t**), indica che in questa fase è stata utilizzata la scansione ritardata. Il valore predefinito è false (**f**).

Query di esempio

Anziché eseguire direttamente la query STV_EXEC_STATE, Amazon Redshift consiglia di eseguire la query SVL_QUERY_SUMMARY o SVV_QUERY_STATE per ottenere le informazioni in STV_EXEC_STATE in un formato più semplice. Per ulteriori dettagli, consultare SVL_QUERY_SUMMARY o SVV_QUERY_STATE.

STV_INFLIGHT

Utilizza la tabella STV_INFLIGHT per determinare quali query sono attualmente in esecuzione sul cluster. Se stai risolvendo problemi, è utile per controllare lo stato delle query di lunga durata.

STV_INFLIGHT non indica query solo di nodo principale. Per ulteriori informazioni, consultare Nodo principale: solo funzioni. STV_INFLIGHT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Risoluzione dei problemi con STV_INFLIGHT

Se utilizzi STV_INFLIGHT per risolvere i problemi relativi alle prestazioni di una query o di una raccolta di query, tieni presente quanto segue:

- Le transazioni aperte a lungo termine generalmente aumentano il carico. Queste transazioni aperte possono comportare tempi di esecuzione più lunghi per altre domande.
- I job COPY ed ETL di lunga durata possono influire su altre query in esecuzione sul cluster, se richiedono molte risorse di elaborazione. Nella maggior parte dei casi, lo spostamento di questi processi di lunga durata in periodi di scarso utilizzo aumenta le prestazioni per i carichi di lavoro di reporting o analisi.
- Esistono visualizzazioni che forniscono informazioni correlate a STV_INFLIGHT. Questi includono STL_QUERYTEXT, che acquisisce il testo della query per i comandi SQL, e SVV_QUERY_INFLIGHT, che unisce STV_INFLIGHT a STL_QUERYTEXT. Puoi anche usare STV_RECENTS con STV_INFLIGHT per la risoluzione dei problemi. Ad esempio, STV_RECENTS può indicare se query specifiche sono nello stato In esecuzione o Completato. La combinazione di queste informazioni con i risultati di STV_INFLIGHT può fornire ulteriori informazioni sulle proprietà di una query e sull'impatto sulle risorse di calcolo.

Puoi anche monitorare le query in esecuzione utilizzando la console Amazon Redshift.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
sezione	integer	Sezione in cui viene eseguita la query.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizzazioni di sistema.
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etiche tta definita con un comando SET QUERY GROUP. Se la query non è basata su file o il parametro QUERY_GROUP non è impostato, questo campo è vuoto.

Nome colonna	Tipo di dati	Descrizione
xid	bigint	ID transazione.
pid	integer	ID processo. Tutte le query in una sessione sono eseguite nello stesso processo, quindi questo valore rimane costante se si esegue una serie di query nella stessa sessione. È possibile utilizzare questa colonna per eseguire la connessione alla tabella STL_ERROR.
starttime	timestamp	Ora in cui è stata avviata la query.
text	character(100)	Testo di query troncato a 100 caratteri se l'istruzione supera quel limite.
sospeso	integer	Indica se la query è sospesa. 0 = false; 1 = true.
insert_pristine	integer	Indica se l'esecuzione delle query di scrittura è/era possibile quando la query corrente è/era in esecuzione. 1 = nessuna query di scrittura consentita. 0 = query di scrittura consentite. Questa colonna è da utilizzarsi per scopi di debug.
concurrency_scaling_status	integer	Indica se la query è stata eseguita nel cluster principale o in un cluster con dimensionamento simultaneo. I valori possibili sono i seguenti: 0 - Eseguita nel cluster principale 1 - Eseguita in un cluster con dimensionamento simultaneo

Query di esempio

Per visualizzare tutte le query attive correntemente in esecuzione sul database, digita la seguente query:

```
select * from stv_inflight;
```

L'output di esempio seguente mostra due query in corso di esecuzione, ovvero la query STV_INFLIGHT stessa e una query eseguita da uno script denominato `avgwait.sql`:

```
select slice, query, trim(label) querylabel, pid,
starttime, substring(text,1,20) querytext
from stv_inflight;

slice|query|querylabel | pid |        starttime         |      querytext
-----+-----+-----------+-----+--------------------------+--------------------
1011 |  21 |           | 646 |2012-01-26 13:23:15.645503|select slice, query,
1011 |  20 |avgwait.sql| 499 |2012-01-26 13:23:14.159912|select avg(datediff(
(2 rows)
```

La seguente query seleziona diverse colonne, tra cui concurrency_scaling_status. Questa colonna indica se le query vengono inviate al cluster con scalabilità concorrenziale. Se il valore è 1 per alcuni risultati, indica che vengono utilizzate risorse di elaborazione con scalabilità concorrenziale. Per ulteriori informazioni, consulta Utilizzo del dimensionamento della simultaneità.

```
select userid,
query,
pid,
starttime,
text,
suspended,
concurrency_scaling_status
 from STV_INFLIGHT;
```

L'output di esempio mostra l'invio di una query al cluster di scalabilità concorrenziale.

```
 query  | pid     |        starttime           |   text                | suspended
 |  concurrency_scaling_status
--------+---------
+----------------------------|------------------------|--------------|-----------------------
1234567 | 123456  | 2012-01-26 13:23:15.645503 | select userid, query...  0
    1
2345678 | 234567  | 2012-01-26 13:23:14.159912 | select avg(datediff(...  0
    0
(2 rows)
```

Per ulteriori suggerimenti sulla risoluzione dei problemi relativi alle prestazioni delle query, vedere Risoluzione dei problemi delle query.

STV_LOAD_STATE

Utilizza la tabella STV_LOAD_STATE per trovare informazioni sullo stato corrente delle istruzioni COPY in corso.

Il comando COPY aggiorna questa tabella dopo ogni milione di record caricati.

STV_LOAD_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
session	integer	PID di sessione del processo che esegue il caricamento.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizzazioni di sistema.
sezione	integer	Numero della sezione di nodo.
pid	integer	ID processo. Tutte le query in una sessione sono eseguite nello stesso processo, quindi questo valore rimane costante se si esegue una serie di query nella stessa sessione.
recordtime	timestamp	Ora in cui il record è stato registrato.
bytes_to_load	bigint	Numero totale di byte che devono essere caricati da questa sezione. È 0 se i dati da caricare sono compressi.
bytes_loaded	bigint	Numero di byte caricati da questa sezione. Se i dati da caricare sono compressi, è il numero di byte caricati dopo la decompressione dei dati.
bytes_to_load_compressed	bigint	Numero totale di byte di dati compressi che devono essere caricati da questa sezione. È 0 se i dati da caricare non sono compressi.

Nome colonna	Tipo di dati	Descrizione
bytes_loaded_compressed	bigint	Numero di byte di dati compressi caricati da questa sezione. È 0 se i dati da caricare non sono compressi.
lines	integer	Numero di linee caricate da questa sezione.
num_files	integer	Numero di file che devono essere caricati da questa sezione.
num_files_complete	integer	Numero di file caricati da questa sezione.
current_file	character (256)	Nome del file che viene caricato da questa sezione.
pct_complete	integer	Percentuale del caricamento di dati completato da questa sezione.

Query di esempio

Per visualizzare l'avanzamento di ogni sezione per un comando COPY, digita la query seguente. Questo esempio utilizza la funzione PG_LAST_COPY_ID() per recuperare informazioni per l'ultimo comando COPY.

```
select slice , bytes_loaded, bytes_to_load , pct_complete from stv_load_state where
 query = pg_last_copy_id();

 slice | bytes_loaded | bytes_to_load | pct_complete
-------+--------------+---------------+--------------
     2 |            0 |             0 |            0
     3 |     12840898 |      39104640 |           32
(2 rows)
```

STV_LOCKS

Utilizza la tabella STV_LOCKS per visualizzare gli aggiornamenti correnti nelle tabelle del database.

Amazon Redshift blocca le tabelle per impedire a due utenti di aggiornare la stessa tabella nello stesso momento. Mentre la tabella STV_LOCKS mostra tutti gli aggiornamenti correnti delle tabelle,

eseguire una query sulla tabella STL_TR_CONFLICT per visualizzare un log dei conflitti di blocco. Utilizza la vista SVV_TRANSACTIONS per identificare le transazioni aperte e i conflitti di blocco.

STV_LOCKS è visibile solo per agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
table_id	bigint	ID di tabella per la tabella che acquisisce il blocco.
last_commit	timestamp	Timestamp per l'ultimo commit nella tabella.
last_update	timestamp	Timestamp per l'ultimo aggiornamento della tabella.
lock_owner	bigint	ID di transazione associato al blocco.
lock_owner_pid	bigint	ID di processo associato al blocco.
lock_owner_start_ts	timestamp	Timestamp per l'ora di inizio della transazione.
lock_owner_end_ts	timestamp	Timestamp per l'ora di fine della transazione.
lock_status	character (22)	Stato del processo in attesa o con blocco.

Query di esempio

Per visualizzare tutti i blocchi nelle transazioni correnti, digita il comando seguente:

```
select table_id, last_update, lock_owner, lock_owner_pid from stv_locks;
```

Questa query restituisce il seguente output di esempio, che visualizza tre blocchi correntemente attivi:

```
 table_id |        last_update         | lock_owner | lock_owner_pid
----------+----------------------------+------------+----------------
```

```
100004  | 2008-12-23 10:08:48.882319 |       1043 |           5656
100003  | 2008-12-23 10:08:48.779543 |       1043 |           5656
100140  | 2008-12-23 10:08:48.021576 |       1043 |           5656
(3 rows)
```

STV_ML_MODEL_INFO

Indicare le informazioni sullo stato corrente del modello di machine learning.

STV_ML_MODEL_INFO è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
schema_name	char(128)	Lo spazio dei nomi del modello.
user_name	char(128)	Il proprietario del modello.
model_name	char(128)	Il nome del modello.
life_cycle	char(20)	Lo stato del ciclo di vita del modello.
is_refreshable	integer	Lo stato del modello se è aggiornabile se le tabelle e le colonne originali nella query di addestramento esistono ancora e l'utente dispone ancora delle relative autorizzazioni. I valori possibili sono: 1 (aggiornabile) e 0 (non aggiornab ile).
model_state	char(128)	Lo stato corrente del modello.

Query di esempio

La query seguente visualizza lo stato corrente dei modelli di machine learning.

```
SELECT schema_name, model_name, model_state
```

```
FROM stv_ml_model_info;

 schema_name |        model_name            |             model_state
-------------+------------------------------+-------------------------------------
 public      | customer_churn_auto_model    | Train Model On SageMaker In Progress
 public      | customer_churn_xgboost_model | Model is Ready
(2 row)
```

STV_MV_DEPS

La tabella STV_MV_DEPS mostra le dipendenze delle viste materializzate su altre viste materializzate in Amazon Redshift.

Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

STV_MV_DEPS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
db_name	char(128)	Il database che contiene la vista materializzata specificata.
schema	char(128)	Lo schema della vista materializzata.
nome	char(128)	Il nome della vista materializzata.
ref_schema	char(128)	Lo schema delle viste materializzate da cui dipende la vista materializzata.
ref_name	char(128)	Il nome della vista materializzata da cui dipende questa vista materializzata.
ref_database_name	char(128)	Il nome del database da cui dipende questa vista materializzata.

Query di esempio

La query seguente restituisce una riga di output che indica che la vista materializzata `mv_over_foo` utilizza la vista materializzata `mv_foo` nella sua definizione come dipendenza.

```
CREATE SCHEMA test_ivm_setup;
CREATE TABLE test_ivm_setup.foo(a INT);
CREATE MATERIALIZED VIEW test_ivm_setup.mv_foo AS SELECT * FROM test_ivm_setup.foo;
CREATE MATERIALIZED VIEW test_ivm_setup.mv_over_foo AS SELECT * FROM
 test_ivm_setup.mv_foo;

SELECT * FROM stv_mv_deps;

 db_name | schema          | name        |   ref_schema   | ref_name |
 ref_database_name
---------+-----------------+-------------+----------------+----------
+------------------
 dev     | test_ivm_setup  | mv_over_foo | test_ivm_setup | mv_foo   | dev
```

STV_MV_INFO

La tabella STV_MV_INFO contiene una riga per ogni visualizzazione materializzata, se i dati sono obsoleti, e le informazioni sullo stato.

Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

STV_MV_INFO è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
db_name	char(128)	Il database che contiene la vista materializzata.
schema	char(128)	Lo schema del database.
nome	char(128)	Il nome della vista materializzata.

Nome colonna	Tipo di dati	Descrizione
updated_upto_xid	bigint	Riservato per uso interno.
is_stale	char(1)	`t` indica che la vista materializzata è obsoleta. In una vista materializzata non aggiornata sono state aggiornate le tabelle di base ma non la vista materializzata. Le informazioni potrebbero non essere accurate se dall'ultimo riavvio non è stato eseguito alcun aggiornamento. La colonna `is_stale` è sempre impostata su `t` se la vista materializzata dipende da una funzione mutabile. Una funzione mutabile restituisce un risultato diverso quando viene fornito lo stesso argomento o gli stessi argomenti. Ad esempio, la maggior parte delle funzioni che restituiscono una data o un timestamp sono funzioni mutabili.
owner_user_name	char(128)	L'utente proprietario della vista materializzata.

Nome colonna	Tipo di dati	Descrizione
stato	integer	Lo stato della vista materializzata come descritto di seguito: • 0: la vista materializzata viene ricalcolata completamente quando viene aggiornata. • 1: la vista materializzata è incrementale. • 101: la vista materializzata non può essere aggiornata a causa di una colonna eliminata. Questo vincolo si applica anche se la colonna non viene utilizzata nella vista materializzata. • 102: la vista materializzata non può essere aggiornata a causa di un tipo di colonna modificato. Questo vincolo si applica anche se la colonna non viene utilizzata nella vista materializzata. • 103: la vista materializzata non può essere aggiornata a causa di una tabella rinominata. • 104: la vista materializzata non può essere aggiornata a causa di una colonna rinominata. Questo vincolo si applica anche se la colonna non viene utilizzata nella vista materializzata. • 105: la vista materializzata non può essere aggiornata a causa di uno schema rinominato.
autorewrite	char(1)	Una `t` indica che la vista materializzata è idonea per la riscrittura automatica delle query.
autorefresh	char(1)	Una `t` indica che la vista materializzata può essere aggiornata automaticamente.

Query di esempio

Per visualizzare lo stato di tutte le viste materializzate, eseguire la query seguente.

```
select * from stv_mv_info;
```

Questa query restituisce l'output di esempio seguente.

```
 db_name |       schema       |   name  | updated_upto_xid | is_stale | owner_user_name
 | state | autorefresh | autorewrite
---------+--------------------+---------+------------------+----------
+-----------------+-------+-------------+------------
 dev     | test_ivm_setup     | mv      |             1031 | f        | catch-22
 |     1 |           1 |           0
 dev     | test_ivm_setup     | old_mv  |              988 | t        | lotr
 |     1 |           0 |           1
```

STV_NODE_STORAGE_CAPACITY

La tabella STV_NODE_STORAGE_CAPACITY mostra i dettagli della capacità di archiviazione totale e della capacità totale utilizzata per ogni nodo di un cluster. Contiene una riga per ogni nodo.

STV_NODE_STORAGE_CAPACITY è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
nodo	integer	Il numero di nodo.
used	integer	Il numero di blocchi del disco da 1 MB correntemente in uso sul nodo. Per i tipi di nodi RA3, i blocchi usati includono sia blocchi memorizzati nella cache locale che blocchi persistenti in Amazon S3.
capacità	integer	La capacità di archiviazione totale di cui è stato eseguito il provisioning per il nodo in blocchi da 1 MB. La capacità include lo spazio riservato da Amazon Redshift su tipi di nodi DC2 per uso interno. La capacità è maggiore della capacità

Nome colonna	Tipo di dati	Descrizione
		nominale del nodo, ovvero la quantità di spazio disponibile per i dati utente. Per i tipi di nodi RA3, questa capacità corrisponde alla quota di archiviazione gestita totale per il cluster. Per ulteriori informazioni sulla capacità per tipo di nodo, consulta Dettagli relativi ai tipi di nodo nella Guida alla gestione di Amazon Redshift.

Query di esempio

> **Note**
>
> I risultati degli esempi seguenti variano in base alle specifiche del nodo del cluster. Aggiungere la colonna `capacity` a SQL SELECT per recuperare la capacità del cluster.

La query seguente restituisce lo spazio utilizzato e la capacità totale in blocchi da 1 MB. Questo esempio è stato eseguito su un cluster dc2.8xlarge a due nodi.

```
select node, used from stv_node_storage_capacity order by node;
```

Questa query restituisce l'output di esempio seguente.

```
 node | used
------+-------
    0 | 30597
    1 | 27089
```

La query seguente restituisce lo spazio utilizzato e la capacità totale in blocchi da 1 MB. Questo esempio è stato eseguito su un cluster ra3.16xlarge a due nodi.

```
select node, used from stv_node_storage_capacity order by node;
```

Questa query restituisce l'output di esempio seguente.

```
 node | used
------+-------
    0 | 30591
    1 | 27103
```

STV_PARTITIONS

Utilizzare la tabella STV_PARTITIONS per determinare le prestazioni relative alla velocità del disco e l'utilizzo del disco per Amazon Redshift.

STV_PARTITIONS contiene una riga per nodo per volume di disco logico.

STV_PARTITIONS è visibile solo per agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
owner	integer	Nodo del disco che possiede la partizione.
host	integer	Nodo che è fisicamente collegato alla partizione.
diskno	integer	Disco contenente la partizione.
part_begin	bigint	Offset della partizione. I dispositivi raw sono partizionati in modo logico per liberare spazio per i blocchi mirror.
part_end	bigint	Fine della partizione.
used	integer	Numero di blocchi del disco da 1 MB correntemente in uso sulla partizione.
tossed	integer	Numero di blocchi pronti per essere eliminati ma che non sono stati ancora rimossi in quanto liberare i relativi indirizzi non è sicuro. Se gli indirizzi fossero liberati immediatamente, una transazione in corso potrebbe scrivere sulla stessa posizione nel disco. Di conseguenza, questi blocchi vengono rilasciati in occasione del commit successivo. I blocchi del

Nome colonna	Tipo di dati	Descrizione
		disco possono essere contrassegnati come "tossed", ad esempio quando una colonna di tabella viene eliminata, durante le operazioni INSERT o durante operazioni di query basate su disco.
capacità	integer	Capacità totale della partizione in blocchi da 1 MB.
reads	bigint	Numero di letture dall'ultimo riavvio del cluster.
writes	bigint	Numero di scritture dall'ultimo riavvio del cluster.
seek_forward	integer	Numero di volte che una richiesta non riguarda l'indirizzo successivo dato l'indirizzo precedente.
seek_back	integer	Numero di volte che una richiesta non riguarda l'indirizzo precedente dato l'indirizzo successivo.
is_san	integer	Se la partizione appartiene a una rete SAN. I valori validi sono **0** (false) o **1** (true).
Non riuscito	integer	Questa colonna è obsoleta.
mbps	integer	Velocità del disco in megabyte al secondo.
mount	character (256)	Percorso di directory al dispositivo.

Query di esempio

La query seguente restituisce lo spazio su disco utilizzato e la capacità in blocchi del disco da 1 MB e calcola l'utilizzo del disco come percentuale dello spazio su disco raw. Questo spazio include lo spazio riservato da Amazon Redshift per uso interno, di conseguenza è superiore alla capacità nominale del disco, ovvero lo spazio su disco disponibile per l'utente. Il parametro Percentuale di spazio su disco utilizzato nella scheda Prestazioni della console di gestione di Amazon Redshift indica la percentuale di capacità nominale del disco utilizzata dal cluster. Ti consigliamo di monitorare il parametro Percentage of Disk Space Used (Percentuale di spazio su disco utilizzata) affinché lo spazio utilizzato non superi la capacità nominale del disco del cluster.

> **Important**
>
> Ti consigliamo vivamente di non superare la capacità nominale del disco del cluster. Sebbene ciò sia possibile in determinate circostanze, il superamento della capacità nominale diminuisce la tolleranza ai guasti del cluster e aumenta il rischio di perdite di dati.

Questo esempio è stato eseguito su un cluster a due nodi con sei partizioni logiche per nodo. Lo spazio dei dischi è utilizzato in modo molto uniforme, con all'incirca il 25% di ogni disco in uso.

```
select owner, host, diskno, used, capacity,
(used-tossed)/capacity::numeric *100 as pctused
from stv_partitions order by owner;

 owner | host | diskno |  used  | capacity | pctused
-------+------+--------+--------+----------+---------
   0   |  0   |    0   | 236480 |  949954  | 24.9
   0   |  0   |    1   | 236420 |  949954  | 24.9
   0   |  0   |    2   | 236440 |  949954  | 24.9
   0   |  1   |    2   | 235150 |  949954  | 24.8
   0   |  1   |    1   | 237100 |  949954  | 25.0
   0   |  1   |    0   | 237090 |  949954  | 25.0
   1   |  1   |    0   | 236310 |  949954  | 24.9
   1   |  1   |    1   | 236300 |  949954  | 24.9
   1   |  1   |    2   | 236320 |  949954  | 24.9
   1   |  0   |    2   | 237910 |  949954  | 25.0
   1   |  0   |    1   | 235640 |  949954  | 24.8
   1   |  0   |    0   | 235380 |  949954  | 24.8

(12 rows)
```

STV_QUERY_METRICS

Contiene informazioni sui parametri, come il numero di righe elaborate, l'uso della CPU, input/output e l'uso del disco, per le query attive in esecuzione nelle code di query definite dall'utente (classi di servizio). Per visualizzare i parametri delle query completate, consultare la tabella di sistema STL_QUERY_METRICS.

I parametri delle query sono campionati a intervalli di un secondo. Di conseguenza, differenti esecuzioni della stessa query potrebbero restituire orari leggermente differenti. Inoltre, è possibile che i segmenti di query che vengono eseguiti in meno di 1 secondo non siano registrati.

STV_QUERY_METRICS traccia e aggrega i parametri a livello di query, segmento e fase. Per ulteriori informazioni sui segmenti e sulle fasi di query, consultare Pianificazione di query e flusso di lavoro di esecuzione. Molti parametri (come `max_rows`, `cpu_time` e così via) sono sommati sulle sezioni di nodo. Per ulteriori informazioni sulle sezioni di nodo, consultare Architettura del sistema di data warehouse.

Per determinare il livello al quale la riga fornisce i parametri, esamina le colonne `segment` e `step_type`:

- Se entrambe le colonne `segment` e `step_type` sono `-1`, la riga fornisce parametri a livello di query.
- Se `segment` non è `-1` e `step_type` è `-1`, la riga fornisce parametri a livello di segmento.
- Se entrambe le colonne `segment` e `step_type` non sono `-1`, la riga fornisce parametri a livello di fase.

STV_QUERY_METRICS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha eseguito la query che ha generato la voce.
service_class	integer	ID della coda di query WLM (classe di servizio). Le code di query sono definite nella configurazione WLM. I parametri sono restituiti solo per le code definite dall'utente.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.

Nome colonna	Tipo di dati	Descrizione
starttime	timestamp	Orario in UTC in cui la query ha avviato l'esecuzione, con 6 cifre di precisione per le frazioni di secondo. Ad esempio: `2009-06-12 11:29:19.131358` .
slices	integer	Numero di sezioni per il cluster.
segment	integer	Numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. I segmenti di query possono essere eseguiti in parallelo. Ogni segmento viene eseguito in un singolo processo. Se il valore del segmento è -1, viene eseguito il rollup dei valori di segmento dei parametri a livello della query.
step_type	integer	Tipo di fase eseguita. Per una descrizione dei tipi di fase, consultare Tipo di fase.
righe	bigint	Numero di righe elaborate da una fase.
max_rows	bigint	Numero massimo di righe prodotte per una fase, aggregate tra tutte le sezioni.
cpu_time	bigint	Tempo CPU utilizzato in microsecondi. A livello del segmento, il tempo CPU totale per il segmento tra tutte le sezioni. A livello della query, la somma del tempo CPU per la query tra tutte le sezioni e tutti i segmenti.
max_cpu_time	bigint	Tempo CPU massimo utilizzato in microsecondi. A livello del segmento, il tempo CPU massimo utilizzato dal segmento tra tutte le sezioni. A livello della query, il tempo CPU massimo utilizzato dal qualsiasi segmento di query.
blocks_read	bigint	Numero di blocchi da 1 MB letti dalla query o dal segmento.

Nome colonna	Tipo di dati	Descrizione
max_blocks_read	bigint	Numero massimo di blocchi da 1 MB letti dal segmento, aggregati tra tutte le sezioni. A livello del segmento, il numero massimo di blocchi da 1 MB letti per il segmento tra tutte le sezioni. A livello della query, il numero massimo di blocchi da 1 MB letti da qualsiasi segmento di query.
run_time	bigint	Tempo di esecuzione totale, sommato tra le sezioni. Il tempo di esecuzione non include il tempo di attesa. A livello del segmento, il tempo di esecuzione per il segmento, sommato tra tutte le sezioni. A livello della query, il tempo di esecuzione per la query sommato tra tutte le sezioni e tutti i segmenti. Poiché questo valore è una somma, il tempo di esecuzione non è correlato al tempo di esecuzione della query.
max_run_time	bigint	Tempo massimo trascorso per un segmento, in microsecondi. A livello del segmento, il tempo di esecuzione massimo per il segmento tra tutte le sezioni. A livello della query, il tempo di esecuzione per qualsiasi segmento di query.
max_blocks_to_disk	bigint	Quantità massima di spazio su disco utilizzata per scrivere risultati intermedi, in blocchi da 1 MB. A livello del segmento, la quantità massima di spazio su disco utilizzato dal segmento tra tutte le sezioni. A livello della query, la quantità massima di spazio su disco utilizzato da qualsiasi segmento di query.
blocks_to_disk	bigint	La quantità di spazio su disco utilizzata da una query o da un segmento per scrivere risultati intermedi, in blocchi da 1 MB.
step	integer	La fase di query eseguita.

Nome colonna	Tipo di dati	Descrizione
max_query_scan_size	bigint	La dimensione massima di dati sottoposti a scansione da una query in MB. A livello del segmento, la quantità massima di dati sottoposti a scansione dal segmento tra tutte le sezioni. A livello della query, la quantità massima di dati sottoposti a scansione da qualsiasi segmento di query.
query_scan_size	bigint	La dimensione dei dati sottoposti a scansione da una query in MB.
query_priority	integer	La priorità della query. I valori possibili sono `-1`, `0`, `1`, `2`, `3` e `4`, dove `-1` indica che la priorità della query non è supportata.
query_queue_time	bigint	La quantità di tempo espressa in microsecondi di permanenza della query nella coda.

Tipo di fase

Nella tabella seguente sono elencati i tipi di fase relativi agli utenti di database. La tabella non elenca i tipi di fasi solo per uso interno. Se il tipo di fase è -1, il parametro non è restituito a livello della fase.

Step type (Tipo di fase)	Descrizione
1	Scansione della tabella
2	Inserimento di righe
3	Aggregazione di righe
6	Fase di ordinamento
7	Fase di merge
8	Fase di distribuzione
9	Fase di diffusione distribuzione
10	Hash join

Step type (Tipo di fase)	Descrizione
11	Merge join
12	Fase di salvataggio
14	Hash
15	Nested loop join
16	Proiezione di campi ed espressioni
17	Limitazione del numero di righe restituite
18	Unique
20	Eliminazione di righe
26	Limitazione del numero di righe ordinate restituite
29	Calcolo di una funzione finestra
32	UDF
33	Unique
37	Restituzione di righe dal nodo principale al client
38	Restituzione di righe dai nodi di calcolo al nodo principale
40	Scansione Spectrum

Query di esempio

Per trovare query attive con un tempo di CPU elevato (più di 1.000 secondi), esegui la query seguente.

```
select query, cpu_time / 1000000 as cpu_seconds
from stv_query_metrics where segment = -1 and cpu_time > 1000000000
order by cpu_time;
```

```
query | cpu_seconds
------+------------
25775 |        9540
```

Per trovare query attive con un nested loop join che hanno restituito più di un milione di righe, esegui la query seguente.

```
select query, rows
from stv_query_metrics
where step_type = 15 and rows > 1000000
order by rows;

query | rows
------+-----------
25775 | 1580225854
```

Per trovare query attive eseguite per più di 60 secondi e con tempo CPU utilizzato inferiore a 10 secondi, esegui la query seguente.

```
select query, run_time/1000000 as run_time_seconds
from stv_query_metrics
where segment = -1 and run_time > 60000000 and cpu_time < 10000000;

query | run_time_seconds
------+-----------------
25775 |              114
```

STV_RECENTS

Utilizza la tabella STV_RECENTS per trovare informazioni sulle query correntemente attive e su quelle eseguite di recente in un database.

STV_RECENTS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Risoluzione dei problemi con STV_RECENTS

STV_RECENTS è particolarmente utile per determinare se una query o una raccolta di query è attualmente in esecuzione o completata. Mostra anche la durata dell'esecuzione di una query. Questo è utile per avere un'idea delle query che richiedono più tempo.

È possibile aggiungere STV_RECENTS ad altre viste di sistema, ad esempio STV_INFLIGHT, per raccogliere metadati aggiuntivi sulle query in esecuzione. C'è un esempio che mostra come eseguire questa operazione nella sezione delle interrogazioni di esempio. Puoi anche utilizzare i record restituiti da questa visualizzazione insieme alle funzionalità di monitoraggio della console Amazon Redshift per la risoluzione dei problemi in tempo reale.

Le viste di sistema complementari a STV_RECENTS includono STL_QUERYTEXT, che recupera il testo della query per i comandi SQL, e SVV_QUERY_INFLIGHT, che unisce STV_INFLIGHT a STL_QUERYTEXT.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
status	character(20)	Stato della query. I valori validi sono **`Running`**, **`Done`**.
starttime	timestamp	Ora in cui è stata avviata la query.
durata	integer	Numero di microsecondi dall'avvio della sessione.
user_name	character(50)	Nome dell'utente che ha eseguito il processo.
db_name	character(50)	Nome del database.
query	character(600)	Testo della query, fino a 600 caratteri. Tutti i caratteri supplementari sono troncati.
pid	integer	ID di processo per la sessione associata alla query, che è sempre -1 per le query completate.

Query di esempio

Per determinare quali query sono attualmente in esecuzione nel database, esegui la seguente query:

```
select user_name, db_name, pid, query
from stv_recents
where status = 'Running';
```

L'output di esempio seguente mostra una singola query in esecuzione nel database TICKIT:

```
user_name | db_name |   pid   | query
----------+---------+---------+-------------
dwuser    | tickit  |  19996  |select venuename, venueseats from
venue where venueseats > 50000 order by venueseats desc;
```

L'esempio seguente restituisce un elenco di query (se esistenti) in esecuzione o nella coda in attesa di essere eseguite:

```
select * from stv_recents where status<>'Done';

status |    starttime        | duration |user_name|db_name| query     | pid
-------+---------------------+----------+---------+-------+-----------+------
Running| 2010-04-21 16:11... | 281566454| dwuser  |tickit | select ...| 23347
```

Questa query non restituisce risultati a meno che non stai eseguendo un certo numero di query simultanee e alcune di queste sono in coda.

L'esempio seguente amplia quello precedente. In questo caso, le query che sono realmente in esecuzione e non in attesa sono escluse dal risultato:

```
select * from stv_recents where status<>'Done'
and pid not in (select pid from stv_inflight);
...
```

Per ulteriori suggerimenti sulla risoluzione dei problemi relativi alle prestazioni delle query, vedere Risoluzione dei problemi delle query.

STV_SESSIONS

Utilizzare la tabella STV_SESSIONS per visualizzare informazioni sulle sessioni utente attive per Amazon Redshift.

Per visualizzare la cronologia delle sessioni, utilizzare la tabella STL_SESSIONS anziché STV_SESSIONS.

STV_SESSIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_SESSION_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
starttime	timestamp	Ora di inizio della sessione.
elaborazione	integer	ID di processo per la sessione.
user_name	character(50)	Utente associato alla sessione.
db_name	character(50)	Nome del database associato alla sessione.
timeout_sec	int	Il tempo massimo in secondi in cui una sessione rimane inattiva o inattiva prima del timeout. 0 indica che non è impostato alcun timeout.

Query di esempio

Per eseguire un controllo rapido e determinare se altri utenti sono correntemente connessi a Amazon Redshift, digitare la seguente query:

```
select count(*)
from stv_sessions;
```

Se il risultato è maggiore di uno, almeno un altro utente è attualmente connesso al database.

Per visualizzare tutte le sessioni attive per Amazon Redshift, digitare la seguente query:

```
select *
from stv_sessions;
```

Il risultato seguente mostra la presenza di quattro sessioni attive correntemente in esecuzione su Amazon Redshift:

```
       starttime          | process |user_name                    | db_name
      | timeout_sec
-------------------------+---------+----------------------------
+---------------------------+-------------
  2018-08-06 08:44:07.50 |   13779 | IAMA:aws_admin:admin_grp    | dev
      | 0
  2008-08-06 08:54:20.50 |   19829 | dwuser                      | dev
      | 120
  2008-08-06 08:56:34.50 |   20279 | dwuser                      | dev
      | 120
  2008-08-06 08:55:00.50 |   19996 | dwuser                      | tickit
      | 0
(3 rows)
```

Il nome utente con prefisso IAMA indica che l'utente ha effettuato l'accesso utilizzando l'accesso Single Sign-On federato. Per ulteriori informazioni, consultare Utilizzo dell'autenticazione IAM per generare credenziali utente di database.

STV_SLICES

Utilizza la tabella STV_SLICES per visualizzare la mappatura corrente di una sezione a un nodo.

Le informazioni in STV_SLICES sono utilizzate principalmente per scopi di analisi.

STV_SLICES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
nodo	integer	Nodo del cluster in cui si trova la sezione.
sezione	integer	Sezione del nodo.
localslice	integer	Queste informazioni sono solo per uso interno.
tipo	character (1)	Queste informazioni sono solo per uso interno.

Query di esempio

Per visualizzare i nodi del cluster che gestiscono le sezioni, digita la query seguente:

```
select node, slice from stv_slices;
```

Questa query restituisce il seguente output di esempio:

```
 node | slice
------+-------
    0 |     2
    0 |     3
    0 |     1
    0 |     0
(4 rows)
```

STV_STARTUP_RECOVERY_STATE

Registra lo stato delle tabelle che sono temporaneamente bloccate durante le operazioni di riavvio del cluster. Amazon Redshift posiziona un blocco temporaneo sulle tabelle durante l'elaborazione delle stesse per risolvere transazioni obsolete dopo un riavvio del cluster.

STV_STARTUP_RECOVERY_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
db_id	integer	ID database.
table_id	integer	ID tabella.
table_name	character(137)	Nome tabella.

Query di esempio

Per determinare quali tabelle sono temporaneamente bloccate, esegui la query seguente dopo un riavvio del cluster.

```
select * from STV_STARTUP_RECOVERY_STATE;

  db_id | tbl_id | table_name
--------+--------+------------
 100044 | 100058 | lineorder
 100044 | 100068 | part
 100044 | 100072 | customer
 100044 | 100192 | supplier
(4 rows)
```

STV_TBL_PERM

La tabella STV_TBL_PERM contiene informazioni sulle tabelle permanenti in Amazon Redshift, incluse le tabelle temporanee create da un utente per la sessione corrente. STV_TBL_PERM contiene informazioni per tutte le tabelle in tutti i database.

Questa tabella differisce da STV_TBL_TRANS, che contiene informazioni sulle tabelle di database transitorie che il sistema crea durante l'elaborazione della query.

STV_TBL_PERM è visibile solo per agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
sezione	integer	Sezione di nodo allocata alla tabella.
id	integer	ID tabella.
nome	character (72)	Nome tabella.
righe	bigint	Numero di righe di dati nella sezione.
sorted_rows	bigint	Numero di righe nella sezione che sono già ordinate sul disco. Se questo numero non corrisponde al numero ROWS, eseguire un vacuum della tabella per riordinare le righe.
temp	integer	Indica se la tabella è una tabella temporanea. 0 = false; 1 = true.
db_id	integer	ID del database in cui è stata creata la tabella.
insert_pristine	integer	Per uso interno.
delete_pristine	integer	Per uso interno.
backup	integer	Valore che indica se la tabella è inclusa negli snapshot del cluster. 0 = no; 1 = yes. Per ulteriori informazioni, consultare il parametro BACKUP per il comando CREATE TABLE.
dist_style	integer	Stile di distribuzione della tabella a cui appartiene la sezione. Per ulteriori informazioni sui valori, consulta Visualizzazione degli stili di distribuzione. Per ulteriori informazioni sugli stili di distribuzione, consulta Stili di distribuzione.
block_count	integer	Numero di blocchi utilizzati dalla sezione. Il valore è -1 quando non è possibile calcolare il numero di blocchi.

Query di esempio

La query seguente restituisce un elenco di ID di tabella e di nomi distinti:

```
select distinct id, name
from stv_tbl_perm order by name;

   id   |          name
--------+------------------------
 100571 | category
 100575 | date
 100580 | event
 100596 | listing
 100003 | padb_config_harvest
 100612 | sales
...
```

Altre tabelle di sistema utilizzano ID di tabella; di conseguenza, sapere quale ID di tabella corrisponde a una determinata tabella può rivelarsi molto utile. In questo esempio, SELECT DISTINCT viene utilizzata per rimuovere i duplicati (le tabelle sono distribuite tra più sezioni).

Per determinare il numero di blocchi utilizzati da ogni colonna nella tabella VENUE, digita la query seguente:

```
select col, count(*)
from stv_blocklist, stv_tbl_perm
where stv_blocklist.tbl = stv_tbl_perm.id
and stv_blocklist.slice = stv_tbl_perm.slice
and stv_tbl_perm.name = 'venue'
group by col
order by col;

 col | count
-----+-------
   0 |     8
   1 |     8
   2 |     8
   3 |     8
   4 |     8
   5 |     8
   6 |     8
   7 |     8
```

```
(8 rows)
```

Note per l'utilizzo

La colonna ROWS include il numero di righe eliminate che non sono state sottoposte a vacuum (o che lo sono state ma con l'opzione SORT ONLY). Di conseguenza, la SUM della colonna ROWS nella tabella STV_TBL_PERM potrebbe non corrispondere al risultato COUNT(*) quando esegui direttamente una query su una determinata tabella. Ad esempio, se 2 righe sono eliminate da VENUE, il risultato COUNT(*) è 200 ma il risultato SUM(ROWS) è ancora 202:

```
delete from venue
where venueid in (1,2);

select count(*) from venue;
count
-------
200
(1 row)

select trim(name) tablename, sum(rows)
from stv_tbl_perm where name='venue' group by name;

tablename | sum
-----------+-----
venue     | 202
(1 row)
```

Per sincronizzare i dati in STV_TBL_PERM, esegui un vacuum completo della tabella VENUE.

```
vacuum venue;

select trim(name) tablename, sum(rows)
from stv_tbl_perm
where name='venue'
group by name;

tablename | sum
-----------+-----
venue     | 200
(1 row)
```

STV_TBL_TRANS

Utilizza la tabella STV_TBL_TRANS per trovare informazioni sulle tabelle di database transitorie attualmente in memoria.

Le tabelle transitorie sono in genere set di righe temporanee utilizzate come risultati intermedi durante l'esecuzione di una query. STV_TBL_TRANS differisce da STV_TBL_PERM in quanto STV_TBL_PERM contiene informazioni sulle tabelle di database permanenti.

STV_TBL_TRANS è visibile solo per agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
sezione	integer	Sezione di nodo allocata alla tabella.
id	integer	ID tabella.
righe	bigint	Numero di righe di dati nella tabella.
formato	bigint	Numero di byte allocati alla tabella.
query_id	bigint	ID query.
ref_cnt	integer	Numero di riferimenti.
from_suspended	integer	Indica se questa tabella è stata creata o meno quando una query era sospesa.
prep_swap	integer	Indica se questa tabella transitoria è preparata o meno per l'eventuale swapping al disco. (lo swapping verrà eseguito in condizioni di memoria insufficiente)

Query di esempio

Per visualizzare le informazioni delle tabelle transitorie per una query con un ID di query di 90, digita il seguente comando:

```
select slice, id, rows, size, query_id, ref_cnt
```

```
from stv_tbl_trans
where query_id = 90;
```

Questa query restituisce le informazioni delle tabelle transitorie per la query 90, come mostrato nel seguente output di esempio:

```
slice | id | rows | size | query_ | ref_ | from_     | prep_
      |    |      |      | id     | cnt  | suspended | swap
------+----+------+------+--------+------+-----------+-------
 1013 | 95 |    0 |    0 |     90 |    4 |         0 |   0
    7 | 96 |    0 |    0 |     90 |    4 |         0 |   0
   10 | 96 |    0 |    0 |     90 |    4 |         0 |   0
   17 | 96 |    0 |    0 |     90 |    4 |         0 |   0
   14 | 96 |    0 |    0 |     90 |    4 |         0 |   0
    3 | 96 |    0 |    0 |     90 |    4 |         0 |   0
 1013 | 99 |    0 |    0 |     90 |    4 |         0 |   0
    9 | 96 |    0 |    0 |     90 |    4 |         0 |   0
    5 | 96 |    0 |    0 |     90 |    4 |         0 |   0
   19 | 96 |    0 |    0 |     90 |    4 |         0 |   0
    2 | 96 |    0 |    0 |     90 |    4 |         0 |   0
 1013 | 98 |    0 |    0 |     90 |    4 |         0 |   0
   13 | 96 |    0 |    0 |     90 |    4 |         0 |   0
    1 | 96 |    0 |    0 |     90 |    4 |         0 |   0
 1013 | 96 |    0 |    0 |     90 |    4 |         0 |   0
    6 | 96 |    0 |    0 |     90 |    4 |         0 |   0
   11 | 96 |    0 |    0 |     90 |    4 |         0 |   0
   15 | 96 |    0 |    0 |     90 |    4 |         0 |   0
   18 | 96 |    0 |    0 |     90 |    4 |         0 |   0
```

In questo esempio, puoi notare che i dati della query sono relativi alle tabelle 95, 96 e 98. Poiché zero byte sono allocati a questa tabella, questa query può essere eseguita nella memoria.

STV_WLM_CLASSIFICATION_CONFIG

Contiene le regole di classificazione correnti per WLM.

STV_WLM_CLASSIFICATION_CONFIG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
id	integer	ID classe di servizio. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.
condition	character(128)	Condizioni di query.
action_seq	integer	Riservato per il sistema.
action	character(64)	Riservato per il sistema.
action_service_class	integer	La classe di servizio in cui viene eseguita l'operazione.

Query di esempio

```
select * from STV_WLM_CLASSIFICATION_CONFIG;

id | condition                                    | action_seq | action |
 action_service_class
---+----------------------------------------------+------------+--------
+---------------------
 1 | (system user) and (query group: health)     |          0 | assign |
     1
 2 | (system user) and (query group: metrics)    |          0 | assign |
     2
 3 | (system user) and (query group: cmstats)    |          0 | assign |
     3
 4 | (system user)                               |          0 | assign |
     4
 5 | (super user) and (query group: superuser)   |          0 | assign |
     5
 6 | (query group: querygroup1)                  |          0 | assign |
     6
 7 | (user group: usergroup1)                    |          0 | assign |
     6
 8 | (user group: usergroup2)                    |          0 | assign |
     7
```

```
 9 | (query group: querygroup3)                  |          0 | assign |
     8
10 | (query group: querygroup4)                  |          0 | assign |
     9
11 | (user group: usergroup4)                    |          0 | assign |
     9
12 | (query group: querygroup*)                  |          0 | assign |
    10
13 | (user group: usergroup*)                    |          0 | assign |
    10
14 | (querytype: any)                            |          0 | assign |
    11
(4 rows)
```

STV_WLM_QMR_CONFIG

Registra la configurazione per le regole di monitoraggio di query WLM (QMR). Per ulteriori informazioni, consultare Regole di monitoraggio delle query WLM.

STV_WLM_QMR_CONFIG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
service_class	integer	ID della coda di query WLM (classe di servizio). Le code di query sono definite nella configurazione WLM. Le regole possono essere definite solo per le code definite dall'uten te. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.
rule_name	character(256)	Nome della regola di monitoraggio di query.
action	character(256)	Operazione della regola. I valori possibili sono `log`, `hop`, `abort` e `change_query_priority` .
metric_name	character(256)	Nome del parametro.
metric_operator	character(256)	Operatore del parametro. I valori possibili sono >, =, <.

Nome colonna	Tipo di dati	Descrizione
metric_value	double	Il valore di soglia per il parametro specificato che avvia un'operazione.
action_value	character(256)	Se `action` è `change_query_priority` , i valori possibili sono `highest`, `high`, `normal`, `low` e `lowest`. Se `action` è `log`, hop o `abort`, il valore è vuoto.

Query di esempio

Per visualizzare le definizioni delle regole QMR per tutte le classi di servizio superiori a 5 (che includono le code definite dall'utente), esegui la query seguente. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.

```
Select *
from stv_wlm_qmr_config
where service_class > 5
order by service_class;
```

STV_WLM_QUERY_QUEUE_STATE

Registra lo stato corrente delle code di query per le classi di servizio.

STV_WLM_QUERY_QUEUE_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
service_class	integer	ID per la classe di servizio. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.
posizione	integer	Posizione della query nella coda. La query con il valore **`position`** più piccolo è la prossima a essere eseguita.
task	integer	ID utilizzato per tenere traccia di una query via il gestore del carico di lavoro. Può essere associato a più ID di query. Se una query viene riavviata, alla stessa viene assegnato un nuovo ID di query ma non un nuovo ID di attività.
query	integer	ID query. Se una query viene riavviata, alla stessa viene assegnato un nuovo ID di query ma non un nuovo ID di attività.
slot_count	integer	Numero di slot di query WLM.
start_time	timestamp	Ora in cui la query è stata inserita nella coda.
queue_time	bigint	Numero di microsecondi durante i quali la query è stata nella coda.

Query di esempio

La query seguente mostra le query nella coda per le classi di servizio superiori a 4.

```
select * from stv_wlm_query_queue_state
where service_class > 4
order by service_class;
```

Questa query restituisce l'output di esempio seguente.

```
 service_class | position | task | query | slot_count |        start_time          |
 queue_time
---------------+----------+------+-------+------------+----------------------------
+------------
             5 |        0 |  455 |   476 |          5 | 2010-10-06 13:18:24.065838 |
 20937257
             6 |        1 |  456 |   478 |          5 | 2010-10-06 13:18:26.652906 |
 18350191
(2 rows)
```

STV_WLM_QUERY_STATE

Registra lo stato corrente delle query monitorate da WLM.

STV_WLM_QUERY_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
xid	integer	ID di transazione della query o della subquery.
task	integer	ID utilizzato per tenere traccia di una query via il gestore del carico di lavoro. Può essere associato a più ID di query. Se una query viene riavviata, alla stessa viene assegnato un nuovo ID di query ma non un nuovo ID di attività.
query	integer	ID query. Se una query viene riavviata, alla stessa viene assegnato un nuovo ID di query ma non un nuovo ID di attività.
service_class	integer	ID per la classe di servizio. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.

Nome colonna	Tipo di dati	Descrizione
slot_count	integer	Numero di slot di query WLM.
wlm_start_time	timestamp	Ora in cui la query è stata inserita nella coda della tabella di sistema o nella coda di query brevi.
stato	character(16)	Stato corrente delle query o della subquery. I valori possibili sono i seguenti: • `Classified` : query è stata assegnata a una classe di servizio. • `Completed` : la query ha terminato l'esecuzione. La query è stata eseguita correttamente o è stata annullata. Per lo stato finale, controlla i risultati di STL_QUERY. • `Dequeued`: solo per uso interno. • `Evicted`: la query è stata rimossa dalla classe di servizio per il riavvio. • `Evicting`: la query viene rimossa dalla classe di servizio per il riavvio. • `Initialized` : solo per uso interno. • `Invalid`: solo per uso interno. • `Queued`: la query è stata inviata alla coda di query perché non erano disponibili slot per eseguirla. • `QueuedWaiting` : la query è in attesa nella coda. • `Rejected`: solo per uso interno. • `Returning` : la query restituisce i risultati al client. • `Running`: la query è in esecuzione. • `TaskAssigned` : solo per uso interno.
queue_time	bigint	Numero di microsecondi durante i quali query è rimasta nella coda.

Nome colonna	Tipo di dati	Descrizione
exec_time	bigint	Numero di microsecondi durante i quali la query è stata in esecuzione.
query_priority	char(20)	La priorità della query. I valori possibili sono `n/a`, `lowest`, `low`, `normal`, `high` e `highest`, dove `n/a` indica che la priorità della query non è supportata.

Query di esempio

La query seguente visualizza tutte le query correntemente in esecuzione nelle classi di servizio superiori a 4. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.

```
select xid, query, trim(state) as state, queue_time, exec_time
from stv_wlm_query_state
where service_class > 4;
```

Questa query restituisce il seguente output di esempio:

```
xid    | query | state   | queue_time | exec_time
-------+-------+---------+------------+-----------
100813 | 25942 | Running |          0 |    1369029
100074 | 25775 | Running |          0 | 2221589242
```

STV_WLM_QUERY_TASK_STATE

Contiene lo stato corrente delle attività di query delle classi di servizio.

STV_WLM_QUERY_TASK_STATE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
service_class	integer	ID per la classe di servizio. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.
task	integer	ID utilizzato per tenere traccia di una query via il gestore del carico di lavoro. Può essere associato a più ID di query. Se una query viene riavviata, alla stessa viene assegnato un nuovo ID di query ma non un nuovo ID di attività.
query	integer	ID query. Se una query viene riavviata, alla stessa viene assegnato un nuovo ID di query ma non un nuovo ID di attività.
slot_count	integer	Numero di slot di query WLM.
start_time	timestamp	Ora di inizio dell'esecuzione della query.
exec_time	bigint	Numero di microsecondi durante i quali la query è rimasta in esecuzione.

Query di esempio

La query seguente visualizza lo stato corrente delle query nelle classi di servizio superiori a 4. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.

```
select * from stv_wlm_query_task_state
where service_class > 4;
```

Questa query restituisce il seguente output di esempio:

```
service_class | task | query |         start_time         | exec_time
--------------+------+-------+----------------------------+-----------
    5         |  466 |   491 | 2010-10-06 13:29:23.063787 | 357618748
(1 row)
```

STV_WLM_SERVICE_CLASS_CONFIG

Registra le configurazioni delle classi di servizio per WLM.

STV_WLM_SERVICE_CLASS_CONFIG è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
service_class	integer	ID per la classe di servizio. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.
queueing_strategy	character(32)	Riservato per il sistema.
num_query_tasks	integer	Livello di simultaneità effettivo corrente della classe di servizio. Se `num_query_tasks` e `target_num_query_tasks` sono differenti, una transizione WLM dinamica è in corso. Un valore `-1` indica che Auto WLM (WLM automatico) è configurato.
target_num_query_tasks	integer	Livello di simultaneità impostato dalla modifica più recente alla configurazione WLM.
evictable	character(8)	Riservato per il sistema.
eviction_threshold	bigint	Riservato per il sistema.
query_working_mem	integer	Quantità effettiva corrente di memoria di lavoro in MB per slot, per nodo, assegnata alla classe di servizio. Se `query_working_mem` e `target_query_working_mem` sono differenti, una transizione WLM dinamica è in corso. Un valore `-1` indica che Auto WLM (WLM automatico) è configurato.
target_query_working_mem	integer	Quantità di memoria di lavoro in MB per slot, per nodo, impostata dalla modifica più recente alla configurazione WLM.

Nome colonna	Tipo di dati	Descrizione
min_step_mem	integer	Riservato per il sistema.
nome	character(64)	Il nome della classe dei servizi.
max_execution_time	bigint	Numero di microsecondi durante i quali la query può essere eseguita prima di essere terminata.
user_group_wild_card	Booleano	Se TRUE, la coda WLM considera un asterisco (*) come carattere jolly nelle stringhe di gruppi di utenti nella configurazione WLM.
query_group_wild_card	Booleano	Se TRUE, la coda WLM considera un asterisco (*) come carattere jolly nelle stringhe di gruppi di query nella configurazione WLM.
concurrency_scaling	character(20)	Descrive se il dimensionamento simultaneo è on o `off`.
query_priority	character(20)	Il valore della priorità della query.
user_role_wild_card	Booleano	Se TRUE, la coda WLM considera un asterisco (*) come carattere jolly nelle stringhe di utenti nella configurazione WLM.

Query di esempio

La prima classe di servizio definita dall'utente è la classe di servizio 6, denominata Classe di servizio n. 1. La seguente query mostra la configurazione corrente per le classi di servizio maggiori di 4. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.

```
select rtrim(name) as name,
num_query_tasks as slots,
query_working_mem as mem,
max_execution_time as max_time,
user_group_wild_card as user_wildcard,
query_group_wild_card as query_wildcard
from stv_wlm_service_class_config
where service_class > 4;
```

```
name                          | slots | mem | max_time | user_wildcard | query_wildcard
------------------------------+-------+-----+----------+---------------+---------------
Service class for super user |     1 | 535 |        0 | false         | false
Queue 1                       |     5 | 125 |        0 | false         | false
Queue 2                       |     5 | 125 |        0 | false         | false
Queue 3                       |     5 | 125 |        0 | false         | false
Queue 4                       |     5 | 627 |        0 | false         | false
Queue 5                       |     5 | 125 |        0 | true          | true
Default queue                 |     5 | 125 |        0 | false         | false
```

La query seguente mostra lo stato di una transizione WLM dinamica. Durante l'elaborazione della transizione, `num_query_tasks` e `target_query_working_mem` vengono aggiornate fino a che non risultano identiche ai valori di destinazione. Per ulteriori informazioni, consultare Proprietà di configurazione dinamiche e statiche WLM.

```
select rtrim(name) as name,
num_query_tasks as slots,
target_num_query_tasks as target_slots,
query_working_mem as memory,
target_query_working_mem as target_memory
from stv_wlm_service_class_config
where num_query_tasks > target_num_query_tasks
or query_working_mem > target_query_working_mem
and service_class > 5;

 name             | slots | target_slots | memory | target_mem
------------------+-------+--------------+--------+------------
 Queue 3          |     5 |           15 |    125 |       375
 Queue 5          |    10 |            5 |    250 |       125
 (2 rows)
```

STV_WLM_SERVICE_CLASS_STATE

Contiene lo stato corrente delle classi di servizio.

STV_WLM_SERVICE_CLASS_STATE è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
service_class	integer	ID per la classe di servizio. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.
num_queued_queries	integer	Numero di query attualmente presenti nella coda.
num_executing_queries	integer	Numero di query attualmente in esecuzione.
num_serviced_queries	integer	Numero di query che in un momento o in un altro erano nella classe di servizio.
num_executed_queries	integer	Numero di query eseguite dal riavvio di Amazon Redshift.
num_evicted_queries	integer	Numero di query rimosse dal riavvio di Amazon Redshift. Alcuni dei motivi della rimozione di una query includono un timeout WLM, un'azione hop QMR e una query non riuscita in un cluster con dimensionamento simultaneo.
num_concurrency_scaling_queries	integer	Numero di query eseguite in un cluster con dimensionamento simultaneo dal riavvio di Amazon Redshift.

Query di esempio

La query seguente visualizza lo stato delle classi di servizio superiori a 5. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.

```
select service_class, num_executing_queries,
num_executed_queries
from stv_wlm_service_class_state
where service_class > 5
order by service_class;
```

```
 service_class | num_executing_queries | num_executed_queries
```

```
--------------+----------------------+---------------------
            6 |                    1 |                 222
            7 |                    0 |                 135
            8 |                    1 |                  39
(3 rows)
```

STV_XRESTORE_ALTER_QUEUE_STATE

Utilizzate STV_XRESTORE_ALTER_QUEUE_STATE per monitorare l'avanzamento della migrazione di ogni tabella durante un ridimensionamento classico. È applicabile in particolare quando il tipo di nodo di destinazione è RA3. Per ulteriori informazioni sul ridimensionamento classico ai nodi RA3, vai a Ridimensionamento classico.

STV_XRESTORE_ALTER_QUEUE_STATE è visibile solo ai superutenti. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_RESTORE_STATE. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha avviato il ridimensionamento.
db_id	integer	L'ID del database.
schema	char(128)	Il nome dello schema.
table_name	char(128)	Nome della tabella.
tbl	integer	L'ID della tabella.
status	char(64)	Lo stato dell'avanzamento della migrazione della tabella. I valori possibili sono i seguenti. • `Waiting`: In attesa dell'inizio della ridistribuzione • `Applying`: Attualmente in fase di ridistribuzione

Nome colonna	Tipo di dati	Descrizione
		• `Finished`: ridistribuzione terminata
task_type	integer	Il tipo di ridistribuzione della tabella. I valori possibili sono i seguenti. • 1: CHIAVE • 2: ANCHE Per ulteriori informazioni sugli stili di distribuzione, vedereStili di distribuzione.

Query di esempio

La seguente query mostra il numero di tabelle in un database che sono in attesa di essere ridimensionate, sono attualmente in fase di ridimensionamento e il ridimensionamento è terminato.

```
select db_id, status, count(*)
from stv_xrestore_alter_queue_state
group by 1,2 order by 3 desc

db_id  |   status   | count
-------+------------+------
694325 | Waiting    |   323
694325 | Finished   |    60
694325 | Applying   |     1
```

Visualizzazioni SVCS per i cluster principale e con scalabilità simultanea

Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e con scalabilità simultanea. Le visualizzazioni sono simili alle tabelle con il prefisso STL, tranne per il fatto che le tabelle STL forniscono informazioni solo per le query eseguite nel cluster principale.

Argomenti

- SVCS_ALERT_EVENT_LOG

- SVCS_COMPILE
- SVCS_CONCURRENCY_SCALING_USAGE
- SVCS_EXPLAIN
- SVCS_PLAN_INFO
- SVCS_QUERY_SUMMARY
- SVCS_S3LIST
- SVCS_S3LOG
- SVCS_S3PARTITION_SUMMARY
- SVCS_S3QUERY_SUMMARY
- SVCS_STREAM_SEGS
- SVCS_UNLOAD_LOG

SVCS_ALERT_EVENT_LOG

Registra un avviso quando il query optimizer identifica delle condizioni che potrebbero indicare problemi di prestazioni. Questa visualizzazione è derivata dalla tabella di sistema STL_ALERT_EVENT_LOG ma non mostra il livello di sezione per le query eseguite in un cluster di dimensionamento della concorrenza. Utilizza la tabella SVCS_ALERT_EVENT_LOG per identificare le possibilità di miglioramento delle performance della query.

Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. Per ulteriori informazioni, consulta Elaborazione query.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili alle tabelle con il prefisso STL, tranne per il fatto che le tabelle STL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_ALERT_EVENT_LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
segment	integer	Numero identificativo del segmento di query.
step	integer	La fase di query eseguita.
pid	integer	ID di processo associato all'istruzione e alla sezione. La stessa query potrebbe avere più PID se viene eseguita su più sezioni.
xid	bigint	ID di transazione associato all'istruzione.
evento	character (1024)	Descrizione dell'evento di avviso.
solution	character (1024)	Soluzione consigliata.
event_time	timestamp	Ora in UTC in cui è stata avviata la query. Il tempo totale include l'inserimento in coda e l'esecuzione, con precisione a 6 cifre per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.131358`** .

Note per l'utilizzo

È possibile utilizzare lo SVCS_ALERT_EVENT_LOG per identificare potenziali problemi nelle query, poi, per ottimizzare la progettazione del database e riscrivere le query, segui le procedure in Ottimizzazione delle prestazioni delle query. SVCS_ALERT_EVENT_LOG registra i seguenti avvisi:

- Statistiche mancanti

Mancano le statistiche. Esegui ANALYZE in seguito a caricamenti di dati o aggiornamenti significativi e utilizza STATUPDATE con le operazioni COPY. Per ulteriori informazioni, consulta Best practice di Amazon Redshift per la progettazione di query.

- Loop nidificato

  Un loop nidificato è solitamente un prodotto cartesiano. Valuta la query per garantire che tutte le tabelle che partecipano siano unite in modo efficiente.

- Filtro molto selettivo

  Il rapporto tra righe restituite e righe scansionate è minore di 0,05. Le righe scansionate corrispondono al valore `rows_pre_user_filter` e le righe restituite al valore delle righe nella tabella di sistema STL_SCAN. Indica che la query sta eseguendo la scansione di un numero insolitamente grande di righe per determinare il set di risultati. Questo può essere dovuto a chiavi di ordinamento mancanti o non corrette. Per ulteriori informazioni, consulta Utilizzo delle chiavi di ordinamento.

- Righe fantasma eccessive

  Una scansione ha ignorato un numero piuttosto grande di righe contrassegnate come cancellate ma non sottoposte a vacuum o righe che sono state inserite ma non eseguite. Per ulteriori informazioni, consulta Vacuum delle tabelle.

- Distribuzione estesa

  Più di 1.000.000 di righe sono state redistribuite per un hash join o un'aggregazione. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati.

- Trasmissione estesa

  Più di 1.000.000 di righe sono state trasmesse per un hash join. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati.

- Esecuzione seriale

  Uno stile di redistribuzione DS_DIST_ALL_INNER è stato indicato nel piano di query, il che forza un'esecuzione seriale perché l'intera tabella interna è stata redistribuita in un solo nodo. Per ulteriori informazioni, consulta Utilizzo degli stili di distribuzione dati.

Query di esempio

La query seguente mostra eventi di avviso per quattro query.

```
SELECT query, substring(event,0,25) as event,
substring(solution,0,25) as solution,
trim(event_time) as event_time from svcs_alert_event_log order by query;

 query |             event             |          solution            |     event_time

-------+-------------------------------+-------------------------------
+---------------------
  6567 | Missing query planner statist | Run the ANALYZE command      | 2014-01-03
 18:20:58
  7450 | Scanned a large number of del | Run the VACUUM command to rec| 2014-01-03
 21:19:31
  8406 | Nested Loop Join in the query | Review the join predicates to| 2014-01-04
 00:34:22
 29512 | Very selective query filter:r | Review the choice of sort key| 2014-01-06
 22:00:00

(4 rows)
```

SVCS_COMPILE

Registra tempo e posizione di compilazione per ogni segmento delle query, comprese le query eseguite in un cluster di dimensionamento e le query eseguite nel cluster principale.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili a quelle con il prefisso SVL, tranne per il fatto che le visualizzazioni SVL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_COMPILE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni su SCL_COMPILE, consultare SVL_COMPILE.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato la voce.
xid	bigint	L'ID di transazione associato all'istruzione.
pid	integer	L'ID di processo associato all'istruzione.
query	integer	L'ID di query. È possibile utilizzare questo ID per unire varie altre tabelle e visualizzazioni di sistema.
segment	integer	Il segmento di query da compilare.
locus	integer	La posizione nel quale è eseguito il segmento. **1** se in un nodo di calcolo e **2** se nel nodo principale.
starttime	timestamp	L'ora in tempo coordinato universale UTC che la compilazione ha avviato.
endtime	timestamp	L'ora in UTC nella quale è stata terminata la compilazione.
compile	integer	Un valore che corrisponde a **0** se la compilazione è stata riutilizzata e a **1** se il segmento è stato compilato.

Query di esempio

In questo esempio, le query 35878 e 35879 hanno eseguito la stessa istruzione SQL. La colonna di compilazione della query 35878 mostra 1 per quattro segmenti di query, cosa che indica che il segmento è stato compilato. La query 35879 mostra `0` nella colonna di compilazione per ogni segmento, cosa che indica che il segmento non deve essere compilato di nuovo.

```
select userid, xid,  pid, query, segment, locus,
datediff(ms, starttime, endtime) as duration, compile
from svcs_compile
where query = 35878 or query = 35879
order by query, segment;
```

```
 userid |  xid   |  pid  | query | segment | locus | duration | compile
--------+--------+-------+-------+---------+-------+----------+---------
    100 | 112780 | 23028 | 35878 |       0 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       1 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       2 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       3 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       4 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       5 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       6 |     1 |     1380 |       1
    100 | 112780 | 23028 | 35878 |       7 |     1 |     1085 |       1
    100 | 112780 | 23028 | 35878 |       8 |     1 |     1197 |       1
    100 | 112780 | 23028 | 35878 |       9 |     2 |      905 |       1
    100 | 112782 | 23028 | 35879 |       0 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       1 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       2 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       3 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       4 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       5 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       6 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       7 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       8 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       9 |     2 |        0 |       0
(20 rows)
```

SVCS_CONCURRENCY_SCALING_USAGE

Registra i periodi di utilizzo del dimensionamento della simultaneità. Ogni periodo di utilizzo è il tempo continuativo impiegato da un singolo cluster di dimensionamento della simultaneità per elaborare attivamente le query.

SVCS_CONCURRENCY_SCALING_USAGE è visibile agli utenti con privilegi avanzati. L'utente con privilegi avanzati del database può scegliere di renderla visibile a tutti gli utenti.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
start_time	timestamp without time zone	Quando inizia il periodo di utilizzo.

Nome colonna	Tipo di dati	Descrizione
end_time	timestamp without time zone	Quando finisce il periodo di utilizzo.
queries	bigint	Numero di query eseguite durante il periodo di utilizzo.
usage_in_seconds	numeric(27,0)	Numero totale di secondi nel periodo di utilizzo.

Query di esempio

Per visualizzare la durata di utilizzo in secondi per un periodo specifico, usa la seguente query:

```
select * from svcs_concurrency_scaling_usage order by start_time;

start_time | end_time | queries | usage_in_seconds
---------------------------+---------------------------+---------+-----------------
2019-02-14 18:43:53.01063 | 2019-02-14 19:16:49.781649 | 48 | 1977
```

SVCS_EXPLAIN

Mostra il piano EXPLAIN per una query inviata in esecuzione.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili alle tabelle con il prefisso STL, tranne per il fatto che le tabelle STL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_EXPLAIN è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
nodeid	integer	Identificatore di nodo di piano, dove un nodo corrisponde a una o più fasi nell'esecuzione della query.
parentid	integer	Identificatore di nodo di piano per un nodo padre. Un nodo padre ha un certo numero di nodi figli. Ad esempio, un merge join è il padre delle scansioni sulle tabelle collegate.
plannode	character(400)	Il testo del nodo dall'output EXPLAIN. I nodi di piano riferiti all'esecuzione sui nodi di calcolo hanno il prefisso **XN** nell'outp ut EXPLAIN.
info	character(400)	Informazioni di qualificatore e di filtro per il nodo di piano. Ad esempio, le condizioni di join e le restrizioni di clausola WHERE sono incluse in questa colonna.

Query di esempio

Considera il seguente output EXPLAIN per una query di join d'aggregazione:

```
explain select avg(datediff(day, listtime, saletime)) as avgwait
from sales, listing where sales.listid = listing.listid;
                                QUERY PLAN

------------------------------------------------------------------------------
 XN Aggregate  (cost=6350.30..6350.31 rows=1 width=16)
  ->  XN Hash Join DS_DIST_NONE  (cost=47.08..6340.89 rows=3766 width=16)
        Hash Cond: ("outer".listid = "inner".listid)
        -> XN Seq Scan on listing  (cost=0.00..1924.97 rows=192497 width=12)
        -> XN Hash  (cost=37.66..37.66 rows=3766 width=12)
              -> XN Seq Scan on sales  (cost=0.00..37.66 rows=3766 width=12)
```

```
(6 rows)
```

Se esegui questa query e l'ID di query è 10, è possibile utilizzare la tabella SVCS_EXPLAIN per visualizzare lo stesso tipo di informazioni restituito dal comando EXPLAIN:

```
select query,nodeid,parentid,substring(plannode from 1 for 30),
substring(info from 1 for 20) from svcs_explain
where query=10 order by 1,2;

query| nodeid |parentid|           substring            |    substring
-----+--------+--------+--------------------------------+-------------------
10   |      1 |      0 |XN Aggregate  (cost=6717.61..6  |
10   |      2 |      1 |  -> XN Merge Join DS_DIST_NO| Merge Cond:("outer"
10   |      3 |      2 |       -> XN Seq Scan on lis |
10   |      4 |      2 |       -> XN Seq Scan on sal |
(4 rows)
```

Considera la query seguente:

```
select event.eventid, sum(pricepaid)
from event, sales
where event.eventid=sales.eventid
group by event.eventid order by 2 desc;

eventid |   sum
--------+----------
    289 | 51846.00
   7895 | 51049.00
   1602 | 50301.00
    851 | 49956.00
   7315 | 49823.00
...
```

Se l'ID di questa query è 15, la seguente query di tabella di sistema restituisce i nodi di piano che sono stati eseguiti. In questo caso, l'ordine dei nodi è invertito per mostrare l'effettivo ordine di esecuzione:

```
select query,nodeid,parentid,substring(plannode from 1 for 56)
from svcs_explain where query=15 order by 1, 2 desc;

query|nodeid|parentid|                          substring
```

```
-----+------+--------+-------------------------------------------------------------
15   |    8 |      7 |                                  -> XN Seq Scan on eve
15   |    7 |      5 |                              -> XN Hash(cost=87.98..87.9
15   |    6 |      5 |                              -> XN Seq Scan on sales(cos
15   |    5 |      4 |                         -> XN Hash Join DS_DIST_OUTER(cos
15   |    4 |      3 |                    -> XN HashAggregate(cost=862286577.07..
15   |    3 |      2 |           -> XN Sort(cost=1000862287175.47..10008622871
15   |    2 |      1 |  -> XN Network(cost=1000862287175.47..1000862287197.
15   |    1 |      0 |XN Merge(cost=1000862287175.47..1000862287197.46 rows=87
(8 rows)
```

La seguente query recupera gli ID di query per tutti i piani di query che contengono una funzione finestra:

```
select query, trim(plannode) from svcs_explain
where plannode like '%Window%';

query|                                      btrim
-----+-------------------------------------------------------------------------
26   | -> XN Window(cost=1000985348268.57..1000985351256.98 rows=170 width=33)
27   | -> XN Window(cost=1000985348268.57..1000985351256.98 rows=170 width=33)
(2 rows)
```

SVCS_PLAN_INFO

Utilizza la tabella SVCS_PLAN_INFO per visualizzare l'output EXPLAIN per una query in termini di un set di righe. Questo è un modo alternativo per visualizzare i piani di query.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili alle tabelle con il prefisso STL, tranne per il fatto che le tabelle STL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_PLAN_INFO è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
nodeid	integer	Identificatore di nodo di piano, dove un nodo corrisponde a una o più fasi nell'esecuzione della query.
segment	integer	Numero identificativo del segmento di query.
step	integer	Numero identificativo della fase di query.
locus	integer	Posizione in cui viene eseguita la fase. 0 se in un nodo di calcolo e 1 se nel nodo principale.
plannode	integer	Valore enumerato del nodo di piano. Consulta la tabella seguente per le enumerazioni del plannode. (La colonna PLANNODE in SVCS_EXPLAIN contiene il testo del nodo di piano.)
startupcost	double precision	Il costo relativo stimato della restituzione della prima riga di questa fase.
totalcost	double precision	Il costo relativo stimato dell'esecuzione della fase.
righe	bigint	Il numero stimato di righe che saranno prodotte dalla fase.
byte	bigint	Il numero stimato di byte che saranno prodotti dalla fase.

Query di esempio

Gli esempi seguenti confrontano i piani di query per una query SELECT semplice restituita utilizzando il comando EXPLAIN e interrogando la tabella SVCS_PLAN_INFO.

```
explain select * from category;
```

```
QUERY PLAN
------------------------------------------------------------
XN Seq Scan on category (cost=0.00..0.11 rows=11 width=49)
(1 row)

select * from category;
catid | catgroup | catname | catdesc
-------+----------+-----------+--------------------------------------------
1 | Sports | MLB | Major League Baseball
3 | Sports | NFL | National Football League
5 | Sports | MLS | Major League Soccer
...

select * from svcs_plan_info where query=256;

query | nodeid | segment | step | locus | plannode | startupcost | totalcost
| rows | bytes
-------+--------+---------+------+-------+----------+-------------+-----------+------
+-------
256 | 1 | 0 | 1 | 0 | 104 | 0 | 0.11 | 11 | 539
256 | 1 | 0 | 0 | 0 | 104 | 0 | 0.11 | 11 | 539
(2 rows)
```

In questo esempio, PLANNODE 104 si riferisce alla scansione sequenziale della tabella CATEGORY.

```
select distinct eventname from event order by 1;

eventname
----------------------------------------------------------------------------
.38 Special
3 Doors Down
70s Soul Jam
A Bronx Tale
...

explain select distinct eventname from event order by 1;

QUERY PLAN
------------------------------------------------------------------------------------------
XN Merge (cost=1000000000136.38..1000000000137.82 rows=576 width=17)
Merge Key: eventname
-> XN Network (cost=1000000000136.38..1000000000137.82 rows=576
width=17)
```

```
Send to leader
-> XN Sort (cost=1000000000136.38..1000000000137.82 rows=576
width=17)
Sort Key: eventname
-> XN Unique (cost=0.00..109.98 rows=576 width=17)
-> XN Seq Scan on event (cost=0.00..87.98 rows=8798
width=17)
(8 rows)

select * from svcs_plan_info where query=240 order by nodeid desc;

query | nodeid | segment | step | locus | plannode | startupcost |
totalcost | rows | bytes
-------+--------+---------+------+-------+----------+-----------------
+------------------+------+--------
240 | 5 | 0 | 0 | 0 | 104 | 0                | 87.98   | 8798 | 149566
240 | 5 | 0 | 1 | 0 | 104 | 0                | 87.98   | 8798 | 149566
240 | 4 | 0 | 2 | 0 | 117 | 0                | 109.975 | 576  | 9792
240 | 4 | 0 | 3 | 0 | 117 | 0                | 109.975 | 576  | 9792
240 | 4 | 1 | 0 | 0 | 117 | 0                | 109.975 | 576  | 9792
240 | 4 | 1 | 1 | 0 | 117 | 0                | 109.975 | 576  | 9792
240 | 3 | 1 | 2 | 0 | 114 | 1000000000136.38 | 1000000000137.82 | 576 | 9792
240 | 3 | 2 | 0 | 0 | 114 | 1000000000136.38 | 1000000000137.82 | 576 | 9792
240 | 2 | 2 | 1 | 0 | 123 | 1000000000136.38 | 1000000000137.82 | 576 | 9792
240 | 1 | 3 | 0 | 0 | 122 | 1000000000136.38 | 1000000000137.82 | 576 | 9792
(10 rows)
```

SVCS_QUERY_SUMMARY

Utilizzare la visualizzazione SVCS_QUERY_SUMMARY per trovare informazioni generali riguardanti l'esecuzione di una query.

Tenere presente che le informazioni in SVCS_QUERY_SUMMARY sono aggregate da tutti i nodi.

> **Note**
>
> La vista SVCS_QUERY_SUMMARY contiene solo informazioni sulle query completate da Amazon Redshift, non su altri comandi di utilità e DDL. Per le informazioni complete e un elenco di tutte le istruzioni completate da Amazon Redshift, compresi i comandi di utilità e DDL, è possibile anche eseguire una query sulla vista SVL_STATEMENTTEXT.
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono

simili a quelle con il prefisso SVL, tranne per il fatto che le visualizzazioni SVL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_QUERY_SUMMARY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Per informazioni su SVL_QUERY_SUMMART, consultare SVL_QUERY_SUMMARY.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizz azioni di sistema.
stm	integer	Flusso: un insieme di segmenti simultanei in una query. Una query ha uno o più flussi.
seg	integer	Numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. I segmenti di query possono essere eseguiti in parallelo. Ogni segmento viene eseguito in un singolo processo.
step	integer	La fase di query eseguita.
maxtime	bigint	Quantità di tempo massima per l'esecuzione della fase (in microsecondi).
avgtime	bigint	Tempo medio per l'esecuzione della fase (in microsecondi).
righe	bigint	Numero di righe di dati coinvolte nella fase di query.

Nome colonna	Tipo di dati	Descrizione
bytes	bigint	Numero di byte di dati coinvolti nella fase di query.
rate_row	double precision	Velocità di esecuzione di query per riga.
rate_byte	double precision	Velocità di esecuzione di query per byte.
etichetta	text	Etichetta di fase, che consiste in un nome di fase di query e, quando applicabile, in un ID di tabella e in un nome di tabella (per esempio, scan tbl=100448 name =user). Gli ID di tabella a tre cifre fanno in genere riferimento alle scansioni delle tabelle transitorie. Quando viene visualizzato `tbl=0`, in genere fa riferimento a una scansione di un valore costante.
is_diskbased	character(1)	Se questa fase della query è stata eseguita come operazione basata su disco su qualsiasi nodo nel cluster: true (**t**) o false (**f**). Solo determinate fasi, come hash, sort e le fasi di aggregazione, possono accedere al disco. Molti tipi di fase sono sempre eseguiti in memoria.
workmem	bigint	Quantità di memoria di lavoro (in byte) assegnata alla fase di query.
is_rrscan	character(1)	Se true (**t**), indica che in questa fase è stata utilizzata la scansione a intervallo limitato. Il valore predefinito è false (**f**).
is_delayed_scan	character(1)	Se true (**t**), indica che in questa fase è stata utilizzata la scansione ritardata. Il valore predefinito è false (**f**).
rows_pre_filter	bigint	Per le scansioni di tabelle permanenti, il numero totale di righe emesse prima di aver applicato filtri alle righe contrassegnate per l'eliminazione (righe fantasma).

Query di esempio

Visualizzazione delle informazioni di elaborazione per una fase di query

La query seguente mostra le informazioni di elaborazione di base per ogni fase della query 87:

```
select query, stm, seg, step, rows, bytes
from svcs_query_summary
where query = 87
order by query, seg, step;
```

Questa query recupera le informazioni di elaborazione relative alla query 87, come mostrato nel seguente output di esempio:

```
 query | stm | seg | step |  rows  |  bytes
-------+-----+-----+------+--------+---------
87     |   0 |   0 |    0 |     90 |    1890
87     |   0 |   0 |    2 |     90 |     360
87     |   0 |   1 |    0 |     90 |     360
87     |   0 |   1 |    2 |     90 |    1440
87     |   1 |   2 |    0 | 210494 | 4209880
87     |   1 |   2 |    3 |  89500 |       0
87     |   1 |   2 |    6 |      4 |      96
87     |   2 |   3 |    0 |      4 |      96
87     |   2 |   3 |    1 |      4 |      96
87     |   2 |   4 |    0 |      4 |      96
87     |   2 |   4 |    1 |      1 |      24
87     |   3 |   5 |    0 |      1 |      24
87     |   3 |   5 |    4 |      0 |       0
(13 rows)
```

Determinare se le fasi della query si sono riversate sul disco

La query seguente mostra se una delle fasi della query con l'ID query 1025 si è riversata o meno sul disco (consultare la visualizzazione SVL_QLOG per scoprire come ottenere l'ID query di una query) o se la query è stata eseguita interamente in memoria:

```
select query, step, rows, workmem, label, is_diskbased
from svcs_query_summary
where query = 1025
order by workmem desc;
```

Questa query restituisce il seguente output di esempio:

```
query| step|  rows  |  workmem   |  label        | is_diskbased
-----+-----+--------+------------+---------------+--------------
```

```
1025 |  0  |16000000|  141557760 |scan tbl=9     | f
1025 |  2  |16000000|  135266304 |hash tbl=142   | t
1025 |  0  |16000000|  128974848 |scan tbl=116536| f
1025 |  2  |16000000|  122683392 |dist           | f
(4 rows)
```

Attraverso la scansione dei valori di IS_DISKBASED, è possibile vedere quali fasi della query sono andate sul disco. Per la query 1025, la fase di hash è stata eseguita su disco. Le fasi che possono essere eseguite su disco includono hash, aggr e le fasi di ordinamento. Per visualizzare solo le fasi di query basate su disco, aggiungere la clausola **`and is_diskbased = 't'`** all'istruzione SQL nell'esempio precedente.

SVCS_S3LIST

Utilizza la vista SVCS_S3LIST per ottenere dettagli sulle query Amazon Redshift Spectrum a livello di segmento. Un segmento può eseguire una scansione della tabella esterna. Questa visualizzazione è derivata dalla visualizzazione di sistema SVL_S3LIST ma non mostra il livello di sezione per le query eseguite in un cluster di dimensionamento della concorrenza.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili a quelle con il prefisso SVL, tranne per il fatto che le visualizzazioni SVL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_S3LIST è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni su SVL_S3LIST, consultare SVL_S3LIST.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query	integer	L'ID di query.

Nome colonna	Tipo di dati	Descrizione
segment	integer	Il numero di segmento. Una query è costituito da più segmenti.
nodo	integer	Il numero di nodo.
eventtime	timestamp	L'orario in formato UTC in cui l'evento viene registrato.
bucket	char(256)	Nome del bucket Amazon S3.
prefisso	char(256)	Il prefisso del posizione del bucket Amazon S3.
recursive	char(1)	Qualora ci sia la scansione ricorsiva delle sottocartelle.
retrieved_files	integer	Il numero di file elencati.
max_file_size	bigint	La dimensione massima dei file tra i file elencati.
vg_file_size	double precision	La dimensione media dei file tra i file elencati.
generated_splits	integer	Il numero di suddivisioni dei file.
avg_split_length	double precision	La lunghezza media delle suddivisioni dei file in byte.
durata	bigint	La durata dell'elenco dei file in microsecondi.

Query di esempio

Le query SVCS_S3LIST di esempio seguenti per l'ultima query eseguita.

```
select *
from svcs_s3list
where query = pg_last_query_id()
```

```
order by query,segment;
```

SVCS_S3LOG

Utilizza la vista SVCS_S3LOG per ottenere informazioni sulla risoluzione di problemi relative alle query Redshift Spectrum a livello di sistema. Un segmento può eseguire una scansione della tabella esterna. Questa visualizzazione è derivata dalla visualizzazione di sistema SVL_S3LOG ma non mostra il livello di sezione per le query eseguite in un cluster di dimensionamento della concorrenza.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili a quelle con il prefisso SVL, tranne per il fatto che le visualizzazioni SVL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_S3LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni su SVL_S3LOG, consultare SVL_S3LOG.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
pid	integer	L'ID di processo.
query	integer	L'ID di query.
segment	integer	Il numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi.
step	integer	La fase di query eseguita.
nodo	integer	Il numero di nodo.
eventtime	timestamp	L'orario in formato UTC in cui l'evento viene registrato.

Nome colonna	Tipo di dati	Descrizione
message	har(512)	Il messaggio per la voce di log.

Query di esempio

Le query SVCS_S3LOG di esempio seguenti per l'ultima query eseguita.

```
select *
from svcs_s3log
where query = pg_last_query_id()
order by query,segment;
```

SVCS_S3PARTITION_SUMMARY

Utilizza la vista SVCS_S3PARTITION_SUMMARY per ottenere un riepilogo dell'elaborazione delle partizioni delle query Redshift Spectrum a livello di segmento. Un segmento può eseguire una scansione della tabella esterna.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili a quelle con il prefisso SVL, tranne per il fatto che le visualizzazioni SVL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_S3PARTITION_SUMMARY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni su SVL_S3PARTITION, consultare SVL_S3PARTITION.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query	integer	L'ID di query. È possibile utilizzare questo valore per unire varie altre tabelle e visualizzazioni di sistema.
segment	integer	Il numero di segmento. Una query è costituito da più segmenti.
incarico	char(1)	Il tipo di incarico di partizione tra nodi.
min_start time	timestamp	L'orario in formato UTC in cui l'elaborazione delle partizion i è iniziata.
max_endti me	timestamp	L'orario in formato UTC in cui l'elaborazione delle partizioni è stata completata.
min_durat ion	bigint	Il periodo minimo di elaborazione delle partizioni utilizzato da un nodo per questa query (in microsecondi).
max_durat ion	bigint	Il periodo massimo di elaborazione delle partizioni utilizzato da un nodo per questa query (in microsecondi).
avg_durat ion	bigint	Il periodo medio di elaborazione delle partizioni utilizzato da un nodo per questa query (in microsecondi).
total_par titions	integer	Il numero totale di partizioni in una tabella esterna.
qualified _partitions	integer	Il numero totale di partizioni qualificate.
min_assig ned_parti tions	integer	Il numero minimo di partizioni assegnate su un nodo.

Nome colonna	Tipo di dati	Descrizione
min_assigned_partitions	integer	Il numero massimo di partizioni assegnate su un nodo.
Il numero minimo di partizioni assegnate su un nodo.	bigint	Il numero medio di partizioni assegnate su un nodo.

Query di esempio

L'esempio seguente dà i dettagli di scansione di partizione per l'ultima query eseguita.

```
select query, segment, assignment, min_starttime, max_endtime, min_duration,
 avg_duration
from svcs_s3partition_summary
where query = pg_last_query_id()
order by query,segment;
```

SVCS_S3QUERY_SUMMARY

Utilizza la vista SVCS_S3QUERY_SUMMARY per ottenere un riepilogo di tutte le query Redshift Spectrum (query S3) che sono state eseguite nel sistema. Un segmento può eseguire una scansione della tabella esterna.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili a quelle con il prefisso SVL, tranne per il fatto che le visualizzazioni SVL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_S3QUERY_SUMMARY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni su SVL_S3QUERY, consultare SVL_S3QUERY.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato quella determinata voce.
query	integer	L'ID di query. È possibile utilizzare questo valore per unire varie altre tabelle e visualizzazioni di sistema.
xid	bigint	L'ID transazione.
pid	integer	L'ID di processo.
segment	integer	Il numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi.
step	integer	La fase di query eseguita.
starttime	timestamp	L'ora in UTC in cui è iniziata l'esecuzione della query Redshift Spectrum in questo segmento. Un segmento può avere una scansione della tabella esterna.
endtime	timestamp	L'ora in UTC in cui è terminata l'esecuzione della query Redshift Spectrum in questo segmento. Un segmento può avere una scansione della tabella esterna.
elapsed	integer	Il tempo impiegato per l'esecuzione della query Redshift Spectrum in questo segmento (in microsecondi).
aborted	integer	Se la query è stata interrotta dal sistema o annullata dall'utente, questa colonna contiene **1**. Se la query è stata completata, questa colonna contiene **0**.

Nome colonna	Tipo di dati	Descrizione
external_table_name	char(136)	Il formato interno del nome esterno della tabella per la scansione della tabella esterna.
file_format	character(16)	Il formato del file dei dati della tabella esterna.
is_partitioned	char(1)	Se true (**t**), questo valore della colonna indica che la tabella esterna è partizionata.
is_rrscan	char(1)	Se true (**t**), questo valore della colonna indica che è stata applicata una scansione a intervallo limitato.
is_nested	varchar(1)	Se true (**t**), questo valore della colonna indica che viene eseguito l'accesso al tipo di dati della colonna nidificata.
s3_scanned_rows	bigint	Il numero di righe di cui è stata eseguita la scansione da Amazon S3 e che sono state inviate al livello Redshift Spectrum.
s3_scanned_bytes	bigint	Il numero di byte di cui è stata eseguita la scansione da Amazon S3 e che sono stati inviati al livello Redshift Spectrum sulla base di dati compressi.
s3query_returned_rows	bigint	Il numero di righe restituite dal livello Redshift Spectrum al cluster.
s3query_returned_bytes	bigint	Il numero di byte restituiti dal livello Redshift Spectrum al cluster. Una grande quantità di dati restituiti a Amazon Redshift può influire sulle prestazioni di sistema.
files	integer	Il numero di file elaborati per questa query Redshift Spectrum. Un piccolo numero di file limita i vantaggi dell'elaborazione parallela.
files_max	integer	Il numero massimo di file elaborati su una sezione.

Nome colonna	Tipo di dati	Descrizione
files_avg	integer	Il numero medio di file elaborati su una sezione.
splits	bigint	Il numero di suddivisioni elaborate per questo segmento. Il numero di suddivisioni elaborate in questa sezione. Con file di dati di grandi dimensioni che possono essere suddivisi, per esempio file di dimensioni superiori a circa 512 MB, Redshift Spectrum cerca di dividere i file in più richieste S3 per l'elaborazione parallela.
splits_max	integer	Il numero massimo di suddivisioni elaborate in questa sezione.
splits_avg	bigint	Il numero medio di suddivisioni elaborate in questa sezione.
total_split_size	bigint	La dimensione totale di tutte le suddivisioni elaborate.
max_split_size	bigint	La dimensione massima di suddivisione elaborata, in byte.
avg_split_size	bigint	La dimensione media di suddivisione elaborata, in byte.
total_retries	bigint	Il numero totale di tentativi per la query Redshift Spectrum in questo segmento.
max_retries	integer	Il numero massimo di tentativi per un singolo file elaborato.
max_request_duration	bigint	La durata massima di una singola richiesta di file (in microsecondi). Le query con un'esecuzione lunga possono indicare un collo di bottiglia.
avg_request_duration	bigint	La durata media delle richieste di file (in microsecondi).

Nome colonna	Tipo di dati	Descrizione
max_request_parallelism	integer	Il numero massimo di richieste parallele in una sezione per questa query Redshift Spectrum.
avg_request_parallelism	double precision	Il numero medio di richieste parallele in una sezione per questa query Redshift Spectrum.
total_slowdown_count	bigint	Il numero totale di richieste Amazon S3 con un errore di rallentamento verificatosi durante la scansione della tabella esterna.
max_slowdown_count	integer	Il numero massimo di richieste di Amazon S3 con un errore di rallentamento verificatosi durante la scansione della tabella esterna in una sezione.

Query di esempio

L'esempio seguente dà i dettagli della fase di scansione per l'ultima query eseguita.

```
select query, segment, elapsed, s3_scanned_rows, s3_scanned_bytes,
 s3query_returned_rows, s3query_returned_bytes, files
from svcs_s3query_summary
where query = pg_last_query_id()
order by query,segment;
```

```
query | segment | elapsed | s3_scanned_rows | s3_scanned_bytes | s3query_returned_rows
 | s3query_returned_bytes | files
------+---------+---------+-----------------+------------------+----------------------
+------------------------+------
 4587 |       2 |   67811 |               0 |                0 |                    0
 |                      0 |     0
 4587 |       2 |  591568 |          172462 |         11260097 |                 8513
 |                 170260 |     1
 4587 |       2 |  216849 |               0 |                0 |                    0
 |                      0 |     0
```

```
 4587 |       2 |  216671 |               0 |                0 |                      0
 |                     0 |     0
```

SVCS_STREAM_SEGS

Elenca la relazione tra flussi e segmenti simultanei.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili alle tabelle con il prefisso STL, tranne per il fatto che le tabelle STL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_STREAM_SEGS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
stream	integer	Il set di segmenti simultanei di una query.
segment	integer	Numero identificativo del segmento di query.

Query di esempio

Per visualizzare la relazione tra flussi e segmenti simultanei per la query più recente, digita la seguente query:

```
select *
```

```
from svcs_stream_segs
where query = pg_last_query_id();

 query | stream | segment
-------+--------+---------
    10 |      1 |       2
    10 |      0 |       0
    10 |      2 |       4
    10 |      1 |       3
    10 |      0 |       1
(5 rows)
```

SVCS_UNLOAD_LOG

Utilizza SVCS_UNLOAD_LOG per ottenere informazioni sulle operazioni UNLOAD.

SVCS_UNLOAD_LOG registra una riga per ogni file creato da un'istruzione UNLOAD. Ad esempio, se un UNLOAD crea 12 file, SVCS_UNLOAD_LOG contiene 12 righe corrispondenti. Questa visualizzazione è derivata dalla tabella di sistema STL_UNLOAD_LOG ma non mostra il livello di sezione per le query eseguite in un cluster di dimensionamento della concorrenza.

> **Note**
>
> Le visualizzazioni di sistema con il prefisso SVCS forniscono i dettagli relativi alle query nei cluster principale e di dimensionamento della simultaneità. Le visualizzazioni sono simili alle tabelle con il prefisso STL, tranne per il fatto che le tabelle STL forniscono informazioni solo per le query eseguite nel cluster principale.

SVCS_UNLOAD_LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato la voce.

Nome colonna	Tipo di dati	Descrizione
query	integer	L'ID di query.
pid	integer	L'ID di processo associato all'istruzione di query.
path	character(1280)	Il percorso completo di un oggetto Amazon S3 per il file.
start_time	timestamp	L'ora di inizio dell'operazione UNLOAD.
end_time	timestamp	L'ora di fine dell'operazione UNLOAD.
line_count	bigint	Il numero di righe scaricate nel file.
transfer_size	bigint	Il numero di byte trasferiti.
file_format	character(10)	Il formato del file scaricato.

Query di esempio

Per ottenere un elenco dei file scritti in Amazon S3 da un comando UNLOAD, puoi chiamare un'operazione di elenco Amazon S3 dopo il completamento di UNLOAD. Tuttavia, in base alla velocità con la quale emetti la chiamata, l'elenco potrebbe essere incompleto perché un'operazione di elenco Amazon S3 è a consistenza finale. Per ottenere immediatamente un elenco completo e autorevole, esegui una query su SVCS_UNLOAD_LOG.

La seguente query restituisce il nome del percorso per i file creati da un UNLOAD per l'ultima query completata:

```
select query, substring(path,0,40) as path
from svcs_unload_log
where query = pg_last_query_id()
order by path;
```

Questo comando restituisce il seguente output di esempio:

```
 query |             path
------+---------------------------------
```

```
  2320 | s3://my-bucket/venue0000_part_00
  2320 | s3://my-bucket/venue0001_part_00
  2320 | s3://my-bucket/venue0002_part_00
  2320 | s3://my-bucket/venue0003_part_00
(4 rows)
```

Viste SVL per il cluster principale

Le viste SVL sono viste di sistema in Amazon Redshift che contengono riferimenti a tabelle e log STL per informazioni più dettagliate.

Queste visualizzazioni forniscono un accesso più rapido e semplice ai dati di query comuni presenti in tali tabelle.

> **Note**
>
> La vista SVL_QUERY_SUMMARY contiene solo informazioni sulle query eseguite da Amazon Redshift, non su altri comandi di utilità e DDL. Per un elenco completo e informazioni su tutte le istruzioni eseguite da Amazon Redshift, inclusi i comandi DDL e di utilità, puoi interrogare la vista SVL_STATEMENTTEXT.

Argomenti

- SVL_AUTO_WORKER_ACTION
- SVL_COMPILE
- SVL_DATASHARE_CHANGE_LOG
- SVL_DATASHARE_CROSS_REGION_USAGE
- SVL_DATASHARE_USAGE_CONSUMER
- SVL_DATASHARE_USAGE_PRODUCER
- SVL_FEDERATED_QUERY
- SVL_MULTI_STATEMENT_VIOLATIONS
- SVL_MV_REFRESH_STATUS
- SVL_QERROR
- SVL_QLOG
- SVL_QUERY_METRICS

- SVL_QUERY_METRICS_SUMMARY
- SVL_QUERY_QUEUE_INFO
- SVL_QUERY_REPORT
- SVL_QUERY_SUMMARY
- SVL_RESTORE_ALTER_TABLE_PROGRESS
- SVL_S3LIST
- SVL_S3LOG
- SVL_S3PARTITION
- SVL_S3PARTITION_SUMMARY
- SVL_S3QUERY
- SVL_S3QUERY_SUMMARY
- SVL_S3RETRIES
- SVL_SPATIAL_SIMPLIFY
- SVL_SPECTRUM_SCAN_ERROR
- SVL_STATEMENTTEXT
- SVL_STORED_PROC_CALL
- SVL_STORED_PROC_MESSAGES
- SVL_TERMINATE
- SVL_UDF_LOG
- SVL_USER_INFO
- SVL_VACUUM_PERCENTAGE

SVL_AUTO_WORKER_ACTION

Registra le azioni automatiche intraprese da Amazon Redshift nelle tabelle definite per l'ottimizzazione automatica.

SVL_AUTO_WORKER_ACTION è visibile a tutti gli utenti. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
table_id	integer	L'identificatore della tabella.
tipo	character (32)	Il tipo di suggerimento. I valori possibili sono distkey e sortkey.
status	character (128)	Lo stato di completamento del suggerimento. I valori possibili sono Start, Complete, Skipped, Abort, Checkpoint e Failed.
eventtime	timestamp	Il timestamp della colonna `status`.
sequenza	integer	Il numero di sequenza di un valroe `previous_state` troncato. Quando una singola `previous_state` contiene più di 200 caratteri , vengono registrate delle righe aggiuntive per tale istruzione. La sequenza è 0 sulla prima riga, 1 sulla seconda e così via.
stato_precedente	character (200)	Lo stile di distribuzione precedente e le chiavi di ordinamento della tabella prima di applicare il suggerimento. Il valore viene troncato in incrementi da 200 caratteri.

Alcuni esempi di valori della colonna `status` sono i seguenti:

- Skipped: tabella non trovata.
- Skipped: il suggerimento è vuoto.
- Skipped: il suggerimento Apply sortkey è disabilitato.
- Skipped: il numero di nuovi tentativi supera il limite massimo per una tabella.
- Skipped: la colonna della tabella è stata modificata.
- Abort: questa tabella non è AUTO.
- Abort: questa tabella è stata convertita di recente.
- Abort: questa tabella supera la soglia delle dimensioni della tabella.
- Abort: questa tabella è già lo stile consigliato.
- Checkpoint: avanzamento *21,9963*%.

Query di esempio

Nell'esempio seguente, le righe del risultato mostrano le azioni intraprese da Amazon Redshift.

```
select table_id, type, status, eventtime, sequence, previous_state
from SVL_AUTO_WORKER_ACTION;
```

```
 table_id |  type   |                        status                        |
 eventtime          | sequence | previous_state
----------+---------+------------------------------------------------------
+----------------------------+----------+----------------
   118082 | sortkey | Start                                                | 2020-08-22
 19:42:20.727049 | 0        |
   118078 | sortkey | Start                                                | 2020-08-22
 19:43:54.728819 | 0        |
   118082 | sortkey | Start                                                | 2020-08-22
 19:42:52.690264 | 0        |
   118072 | sortkey | Start                                                | 2020-08-22
 19:44:14.793572 | 0        |
   118082 | sortkey | Failed                                               | 2020-08-22
 19:42:20.728917 | 0        |
   118078 | sortkey | Complete                                             | 2020-08-22
 19:43:54.792705 | 0        | SORTKEY: None;
   118086 | sortkey | Complete                                             | 2020-08-22
 19:42:00.72635  | 0        | SORTKEY: None;
   118082 | sortkey | Complete                                             | 2020-08-22
 19:43:34.728144 | 0        | SORTKEY: None;
   118072 | sortkey | Skipped:Retry exceeds the maximum limit for a table. | 2020-08-22
 19:44:46.706155 | 0        |
   118086 | sortkey | Start                                                | 2020-08-22
 19:42:00.685255 | 0        |
   118082 | sortkey | Start                                                | 2020-08-22
 19:43:34.69531  | 0        |
   118072 | sortkey | Start                                                | 2020-08-22
 19:44:46.703331 | 0        |
   118082 | sortkey | Checkpoint: progress 14.755079%                      | 2020-08-22
 19:42:52.692828 | 0        |
   118072 | sortkey | Failed                                               | 2020-08-22
 19:44:14.796071 | 0        |
   116723 | sortkey | Abort:This table is not AUTO.                        | 2020-10-28
 05:12:58.479233 | 0        |
```

```
  110203 | distkey | Abort:This table is not AUTO.                    | 2020-10-28
05:45:54.67259  | 0       |
```

SVL_COMPILE

I record compilano l'ora e la posizione per ogni segmento di query delle query.

SVL_COMPILE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> SVL_COMPILE contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Per informazioni su SVCS_COMPILE, consultare SVCS_COMPILE.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
xid	bigint	ID di transazione associato all'istruzione.
pid	integer	ID di processo associato all'istruzione.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizzazioni di sistema.
segment	integer	Il segmento di query da compilare.
locus	integer	La posizione nel quale è eseguito il segmento. **1** se in un nodo di calcolo e **2** se nel nodo principale.

Nome colonna	Tipo di dati	Descrizione
starttime	timestamp	Ora in UTC in cui è stata avviata la compilazione.
endtime	timestamp	Ora in UTC in cui è stata terminata la compilazione.
compile	integer	**0** se la compilazione è stata riutilizzata, **1** se il segmento è stato compilato.

Query di esempio

In questo esempio, le query 35878 e 35879 hanno eseguito la stessa istruzione SQL. La colonna di compilazione della query 35878 mostra 1 per quattro segmenti di query, cosa che indica che il segmento è stato compilato. La query 35879 mostra `0` nella colonna di compilazione per ogni segmento, cosa che indica che il segmento non deve essere compilato di nuovo.

```
select userid, xid,  pid, query, segment, locus,
datediff(ms, starttime, endtime) as duration, compile
from svl_compile
where query = 35878 or query = 35879
order by query, segment;

 userid |  xid   |  pid  | query | segment | locus | duration | compile
--------+--------+-------+-------+---------+-------+----------+---------
    100 | 112780 | 23028 | 35878 |       0 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       1 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       2 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       3 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       4 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       5 |     1 |        0 |       0
    100 | 112780 | 23028 | 35878 |       6 |     1 |     1380 |       1
    100 | 112780 | 23028 | 35878 |       7 |     1 |     1085 |       1
    100 | 112780 | 23028 | 35878 |       8 |     1 |     1197 |       1
    100 | 112780 | 23028 | 35878 |       9 |     2 |      905 |       1
    100 | 112782 | 23028 | 35879 |       0 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       1 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       2 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       3 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       4 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       5 |     1 |        0 |       0
```

```
    100 | 112782 | 23028 | 35879 |       6 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       7 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       8 |     1 |        0 |       0
    100 | 112782 | 23028 | 35879 |       9 |     2 |        0 |       0
(20 rows)
```

SVL_DATASHARE_CHANGE_LOG

Registra la vista consolidata per tenere traccia delle modifiche apportate alle unità di condivisione dati nei cluster di produttori e consumer.

SVL_DATASHARE_CHANGE_LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_DATASHARE_CHANGE_LOG. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che esegue l'operazione.
username	varchar(128)	Il nome dell'utente che esegue l'operazione.
pid	integer	L'ID del processo.
xid	bigint	L'ID della transazione.
share_id	integer	L'ID dell'unità di condivisione dati è interessato.
share_name	varchar(128)	Il nome dell'unità di condivisione dati.
source_database_id	integer	L'ID del database a cui appartiene l'unità di condivisione dati.

Nome colonna	Tipo di dati	Descrizione
source_database_name	varchar(128)	Il nome del database a cui appartiene l'unità di condivisione dati.
consumer_database_id	integer	ID del database importato dall'unità di condivisione dati.
consumer_database_name	varchar(128)	Il nome del database importato dall'unità di condivisione dati.
arn	varchar(192)	L'ARN della risorsa che supporta il database importato.
recordtime	timestamp	Il timestamp dell'operazione.
action	varchar(128)	L'operazione in esecuzione. I valori possibili sono CREATE DATASHARE, DROP DATASHARE, GRANT ALTER, REVOKE ALTER, GRANT SHARE, REVOKE SHARE, ALTER ADD, ALTER REMOVE, ALTER SET, GRANT USAGE, REVOKE USAGE, CREATE DATABASE, GRANT o REVOKE USAGE su un database condiviso, DROP SHARED DATABASE, ALTER SHARED DATABASE.
status	integer	Lo stato dell'operazione. I valori possibili sono SUCCESS e ERROR-ERROR CODE
share_object_type	varchar(64)	Il tipo di oggetto di database aggiunto o rimosso dall'unità di condivisione dati. I valori possibili sono schemi, tabelle, colonne, funzioni e viste. Questo è un campo per il cluster di produttori.
share_object_id	integer	L'ID dell'oggetto di database aggiunto o rimosso dall'unità di condivisione dati. Questo è un campo per il cluster di produttori.
share_object_name	varchar(128)	Il nome dell'oggetto di database aggiunto o rimosso dall'unità di condivisione dati. Questo è un campo per il cluster di produttori.

Nome colonna	Tipo di dati	Descrizione
target_user_type	varchar(16)	Il tipo di utenti o gruppi a cui è stato concesso un privilegio. Questo è un campo per il cluster di produttori e consumer.
target_userid	integer	L'ID di utenti o gruppi a cui è stato concesso un privilegio. Questo è un campo per il cluster di produttori e consumer.
target_username	varchar(128)	Il nome di utenti o gruppi a cui è stato concesso un privilegio. Questo è un campo per il cluster di produttori e consumer.
consumer_account	varchar(16)	L'ID account del consumer di dati. Questo è un campo per il cluster di produttori.
consumer_namespace	varchar(64)	Lo spazio dei nomi dell'account del consumer di dati. Questo è un campo per il cluster di produttori.
producer_account	varchar(16)	L'ID account dell'account del produttore a cui appartiene l'unità di condivisione dati. Questo è un campo per il cluster di consumer.
producer_namespace	varchar(64)	Lo spazio dei nomi dell'account di produzione a cui appartiene l'unità di condivisione dati. Questo è un campo per il cluster di consumer.
attribute_name	varchar(64)	Il nome di un attributo dell'unità di condivisione dati o del database condiviso.
attribute_value	varchar(128)	Il valore di un attributo dell'unità di condivisione dati o del database condiviso.
message	varchar(512)	Il messaggio di errore visualizzato quando un'operazione non riesce.

Query di esempio

L'esempio seguente mostra una vista SVL_DATASHARE_CHANGE_LOG.

```
SELECT DISTINCT action
FROM svl_datashare_change_log
WHERE share_object_name LIKE 'tickit%';
```

```
        action
-----------------------
 "ALTER DATASHARE ADD"
```

SVL_DATASHARE_CROSS_REGION_USAGE

Utilizza la vista SVL_DATASHARE_CROSS_REGION_USAGE per ottenere un riepilogo dell'utilizzo dei dati trasferiti tra regioni causato da query dell'unità di condivisione dati tra regioni. SVL_DATASHARE_CROSS_REGION_USAGE aggrega i dettagli a livello di segmento.

SVL_DATASHARE_CROSS_REGION_USAGE è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_DATASHARE_CROSS_REGION_USAGE. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query	integer	L'ID della query. Utilizza questo valore per unire altre tabelle e viste di sistema.
segment	bigint	Il numero del segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi.
start_time	time	L'ora in UTC in cui è iniziato il trasferimento dei dati.
end_time	time	L'ora in UTC in cui è terminato il trasferimento dei dati.
transferred_data	bigint	Il numero di byte di dati trasferiti da una regione producer a una regione consumer.
source_region	char(25)	La regione producer da cui la query ha trasferito i dati.

Nome colonna	Tipo di dati	Descrizione
recordtime	timestamp	L'ora in cui viene registrata l'operazione.

Query di esempio

L'esempio seguente mostra una vista SVL_DATASHARE_CROSS_REGION_USAGE.

```
SELECT query, segment, transferred_data, source_region
from svl_datashare_cross_region_usage
where query = pg_last_query_id()
order by query,segment;

  query | segment | transferred_data | source_region
--------+---------+------------------+---------------
 200048 |       2 |          4194304 |    us-west-1
 200048 |       2 |          4194304 |    us-east-2
```

SVL_DATASHARE_USAGE_CONSUMER

Registra l'attività e l'utilizzo delle unità di condivisione dati. Questa visualizzazione è rilevante solo per il cluster di consumer.

SVL_DATASHARE_USAGE_CONSUMER è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_DATASHARE_USAGE_CONSUMER. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che invia la richiesta.

Nome colonna	Tipo di dati	Descrizione
pid	integer	L'ID del processo leader che esegue la query.
xid	bigint	Il contesto della transazione corrente.
request_id	varchar(50)	L'ID univoco della chiamata API richiesta.
request_type	varchar(25)	Il tipo di richiesta presentata al cluster di produttori.
transaction_uid	varchar(50)	L'ID univoco della transazione.
recordtime	timestamp	L'ora in cui viene registrata l'operazione.
status	integer	Lo stato della chiamata API richiesta.
error	varchar(512)	Il messaggio per un errore.

Query di esempio

L'esempio seguente mostra una vista SVL_DATASHARE_USAGE_CONSUMER.

```
SELECT request_type, status, trim(error) AS error
FROM svl_datashare_usage_consumer

  request_type  | status | error
----------------+--------+--------
 "GET RELATION" |   0    |
```

SVL_DATASHARE_USAGE_PRODUCER

Registra l'attività e l'utilizzo delle unità di condivisione dati. Questa visualizzazione è rilevante solo per il cluster di consumer.

SVL_DATASHARE_USAGE_PRODUCER è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_DATASHARE_USAGE_PRODUCER. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
share_id	integer	L'ID oggetto (OID) dell'unità di condivisione dati.
share_name	varchar(128)	Il nome dell'unità di condivisione dati.
request_id	varchar(50)	L'ID univoco della chiamata API richiesta.
request_type	varchar(25)	Il tipo di richiesta presentata al cluster di produttori.
object_type	varchar(64)	Il tipo di oggetto condiviso dall'unità di condivisione dati. I valori possibili sono schemi, tabelle, colonne, funzioni e viste.
object_oid	integer	L'ID oggetto condiviso dall'unità di condivisione dati.
object_name	varchar(128)	Il nome dell'oggetto condiviso dall'unità di condivisione dati.
consumer_account	varchar(16)	L'account dell'account consumer con cui è condivisa l'unità di condivisione dati.
consumer_namespace	varchar(64)	Lo spazio dei nomi dell'account consumer con cui è condivisa l'unità di condivisione dati.

Nome colonna	Tipo di dati	Descrizione
consumer_transaction_uid	varchar(50)	L'ID transazione univoco dell'istruzione nel cluster di consumer.
recordtime	timestamp	L'ora in cui viene registrata l'operazione.
status	integer	Lo stato dell'unità di condivisione dati.
error	varchar(512)	Il messaggio per un errore.
consumer_region	char(64)	La regione in cui si trova il cluster consumer.

Query di esempio

L'esempio seguente mostra una vista SVL_DATASHARE_USAGE_PRODUCER.

```
SELECT DISTINCT request_type
FROM svl_datashare_usage_producer
WHERE object_name LIKE 'tickit%';

   request_type
 ------------------
   "GET RELATION"
```

SVL_FEDERATED_QUERY

Utilizzare la visualizzazione SVL_FEDERATED_QUERY per visualizzare informazioni su una chiamata di query federata.

SVL_FEDERATED_QUERY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_EXTERNAL_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da

essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che esegue la query.
xid	bigint	L'ID transazione.
pid	integer	L'ID del processo leader che esegue la query.
query	integer	L'ID di query di una chiamata federata.
sourcetype	carattere (32)	Il tipo di origine della chiamata federata, ad esempio "PG".
recordtime	timestamp	L'ora in cui viene inviata una query per la federazione. Viene utilizzato UTC.
querytext	carattere (4000)	La stringa di query inviata al motore PostgreSQL remoto per l'esecuzione.
num_rows	bigint	Il numero di righe restituito dalla query federata.
num_bytes	bigint	Il numero di byte restituito dalla query federata.
durata	bigint	Il tempo (in microsecondi) dedicato al recupero delle righe dalle chiamate del cursore. Il tempo impiegato

Nome colonna	Tipo di dati	Descrizione
		per eseguire la query federata, oltre a recuperare i risultati.

Query di esempio

Per visualizzare informazioni sulle chiamate di query federate, eseguire la seguente query.

```
select query, trim(sourcetype) as type, recordtime, trim(querytext) as "PG Subquery"
 from svl_federated_query where query = 4292;

 query | type |         recordtime         |                          pg subquery
-------+------+----------------------------
+-------------------------------------------------------------
  4292 | PG   | 2020-03-27 04:29:58.485126 | SELECT "level" FROM functional.employees
 WHERE ("level" >= 6)
(1 row)
```

SVL_MULTI_STATEMENT_VIOLATIONS

Utilizzare la visualizzazione SVL_MULTI_STATEMENT_VIOLATIONS per ottenere un record completo di tutti i comandi SQL eseguiti nel sistema che viola le restrizioni dei blocchi di transazioni.

Le violazioni si verificano quando emetti uno dei seguenti comandi SQL che Amazon Redshift limita all'interno di un blocco di transazioni o di richieste con più istruzioni:

- CREATE DATABASE
- DROP DATABASE
- ALTER TABLE APPEND
- CREATE EXTERNAL TABLE
- DROP EXTERNAL TABLE
- RENAME EXTERNAL TABLE
- ALTER EXTERNAL TABLE
- CREATE TABLESPACE
- DROP TABLESPACE
- CREATE LIBRARY

- DROP LIBRARY
- REBUILDCAT
- INDEXCAT
- REINDEX DATABASE
- VACUUM
- GRANT
- COPY

> **Note**
>
> Se sono presenti voci in questa visualizzazione, allora sarà necessario modificare le applicazioni e gli script SQL corrispondenti. Si consiglia di modificare il codice dell'applicazione per spostare l'utilizzo di questi comandi SQL limitati al di fuori del blocco delle transazioni. Se hai bisogno di ulteriore assistenza, contatta l' AWS assistenza.

SVL_MULTI_STATEMENT_VIOLATIONS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha causato la violazione.
database	character(32)	Il nome del database a cui era connesso l'utente.
cmdname	character(20)	Il nome del comando che non può essere eseguito all'interno di un blocco di transazione o di una richiesta con più istruzioni. Ad esempio, CREATE DATABASE, DROP DATABASE, ALTER TABLE APPEND, CREATE

Nome colonna	Tipo di dati	Descrizione
		EXTERNAL TABLE, DROP EXTERNAL TABLE, RENAME EXTERNAL TABLE, ALTER EXTERNAL TABLE, CREATE LIBRARY, DROP LIBRARY, REBUILDCAT, INDEXCAT, REINDEX DATABASE, VACUUM, GRANT sulle risorse esterne, CLUSTER, COPY, CREATE TABLESPACE e DROP TABLESPACE.
xid	bigint	L'ID di transazione associato all'istruzione.
pid	integer	L'ID di processo per l'istruzione.
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etiche tta definita con un comando SET QUERY GROUP. Se la query non è basata su file o il parametro QUERY_GROUP non è impostato, questo campo è vuoto.
starttime	timestamp	L'ora esatta in cui è iniziata l'esecuzione dell'istruzione, con 6 cifre di precisione per le frazioni di secondo, ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'ora esatta in cui è terminata l'esecuzione dell'istruzione, con 6 cifre di precisione per le frazioni di secondo, ad esempio: **`2009-06-12 11:29:19.193640`** .
sequenza	integer	Quando una singola istruzione contiene più di 200 caratteri , vengono registrate delle righe aggiuntive per tale istruzion e. La sequenza 0 è la prima riga, 1 la seconda e così via.
tipo	varchar(10)	Il tipo di istruzione SQL: **`QUERY`**, **`DDL`** o **`UTILITY`**.
text	character(200)	Il testo SQL, in incrementi da 200 caratteri. Questo campo potrebbe contenere caratteri speciali come barra rovesciat a (\\) e nuova riga (\n).

Query di esempio

La query seguente restituisce più istruzioni che hanno delle violazioni.

```
select * from svl_multi_statement_violations order by starttime asc;

userid | database | cmdname | xid | pid | label | starttime | endtime | sequence | type
 | text
================================================================================================
1 | dev | CREATE DATABASE | 1034 | 5729 |label1 | ******** | ******* | 0 | DDL |
 create table c(b int);
1 | dev | CREATE DATABASE | 1034 | 5729 |label1 | ******** | ******* | 0 | UTILITY |
 create database b;
1 | dev | CREATE DATABASE | 1034 | 5729 |label1 | ******** | ******* | 0 | UTILITY |
 COMMIT
...
```

SVL_MV_REFRESH_STATUS

La vista SVL_MV_REFRESH_STATUS contiene una riga per l'attività di aggiornamento delle viste personalizzate.

Per ulteriori informazioni sulle viste materializzate, consultare Creazione di viste materializzate in Amazon Redshift.

SVL_MV_REFRESH_STATUS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_MV_REFRESH_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
db_name	char(128)	Il database che contiene la vista materializzata.
userid	bigint	L'ID dell'utente che ha eseguito l'aggiornamento.

Nome colonna	Tipo di dati	Descrizione
schema_name	char(128)	Lo schema della vista materializzata.
mv_name	char(128)	Il nome della vista materializzata.
xid	bigint	L'ID della transazione dell'aggiornamento.
starttime	timestamp	L'ora di inizio dell'aggiornamento.
endtime	timestamp	L'ora di fine dell'aggiornamento.

Nome colonna	Tipo di dati	Descrizione
status	text	Lo stato della richiesta. Esempi di valori possibili sono: • Refresh successfully updated MV incrementally (Aggiornamento MV aggiornato correttamente in modo incrementale) Se si tratta di una vista materializzata per lo streaming, il messaggio può contenere ulteriori qualificatori relativi al numero di record. Questi sono i seguenti: • Stream returned no new data (Il flusso non ha restituito nuovi dati): non sono stati recuperat i record. • Tutti i record ricevuti dal flusso sono stati ignorati: i record sono stati recuperati, ma a causa di un errore sono stati tutti ignorati. • Alcuni record del flusso sono stati ignorati: i record sono stati recuperati, ma a causa di un errore alcuni sono stati ignorati. Se non sono presenti qualificatori, è stato recuperato almeno un record e tutti i record sono disponibili nella vista materializzata. Resta un qualificatore possibile: • The stream may contain more data (il flusso può contenere altri dati): l'aggiornamento è terminato prima che Amazon Redshift determinasse che non c'erano altri record da utilizzare. Il flusso può essere aggiornato, ma non è stato confermato da Amazon Redshift. • Refresh successfully recomputed MV from scratch (Aggiornamento MV ricalcolato correttamente da zero)

Nome colonna	Tipo di dati	Descrizione
		• Refresh partially updated MV incrementally up to an active transaction (Aggiornamento MV aggiornato parzialmente in modo incrementale fino a una transazione attiva) • MV was already updated (MV è già stato aggiornato) • Refresh failed. (Aggiornamento non riuscito) A base table column was renamed (Una colonna della tabella di base è stata rinominata) • Refresh failed. (Aggiornamento non riuscito) A base table column type was changed (Un tipo di colonna della tabella di base è stato modificato) • Refresh failed. (Aggiornamento non riuscito) A base table was renamed (Una tabella di base è stata rinominata) • Refresh failed due to an internal error (Aggiornamento non riuscito a causa di un errore interno) • Refresh failed. (Aggiornamento non riuscito) A base table column was dropped (Una colonna della tabella di base è stata eliminata) • Refresh failed. (Aggiornamento non riuscito) Schema of MV was renamed (Schema di MV rinominato) • Refresh failed. (Aggiornamento non riuscito) MV was not found (MV non trovato) • Auto refresh aborted due to excessive user workload (Aggiornamento automatico interrotto a causa di eccessivo carico di lavoro dell'utente) • Refresh failed. (Aggiornamento non riuscito) Serializable isolation violation (Violazione di isolamento serializzabile)

Nome colonna	Tipo di dati	Descrizione
refresh_type	char(32)	La definizione del tipo di aggiornamento. I valori di esempio includono Manuale e Automatico.

Query di esempio

Per visualizzare lo stato di aggiornamento delle viste materializzate, eseguire la query seguente.

```
select * from svl_mv_refresh_status;
```

Questa query restituisce il seguente output di esempio:

```
 db_name | userid |  schema   |  name   |  xid  |         starttime          |
       endtime           |                        status                        |
 refresh_type
---------+--------+-----------+---------+-------+----------------------------
+----------------------------+------------------------------------------------------
+-----------------
 dev     |    169 | mv_schema | mv_test |  6640 | 2020-02-14 02:26:53.497935 |
 2020-02-14 02:26:53.556156 | Refresh successfully recomputed MV from scratch     |
 Manual
 dev     |    166 | mv_schema | mv_test |  6517 | 2020-02-14 02:26:39.287438 |
 2020-02-14 02:26:39.349539 | Refresh successfully updated MV incrementally       |
 Auto
 dev     |    162 | mv_schema | mv_test |  6388 | 2020-02-14 02:26:27.863426 |
 2020-02-14 02:26:27.918307 | Refresh successfully recomputed MV from scratch     |
 Manual
 dev     |    161 | mv_schema | mv_test |  6323 | 2020-02-14 02:26:20.020717 |
 2020-02-14 02:26:20.080002 | Refresh successfully updated MV incrementally       |
 Auto
 dev     |    161 | mv_schema | mv_test |  6301 | 2020-02-14 02:26:05.796146 |
 2020-02-14 02:26:07.853986 | Refresh successfully recomputed MV from scratch     |
 Manual
 dev     |    153 | mv_schema | mv_test |  6024 | 2020-02-14 02:25:18.762335 |
 2020-02-14 02:25:20.043462 | MV was already updated                              |
 Manual
 dev     |    143 | mv_schema | mv_test |  5557 | 2020-02-14 02:24:23.100601 |
 2020-02-14 02:24:23.100633 | MV was already updated                              |
 Manual
```

```
dev     |    141 | mv_schema | mv_test |  5447 | 2020-02-14 02:23:54.102837 |
2020-02-14 02:24:00.310166 | Refresh successfully updated MV incrementally       |
Auto
dev     |      1 | mv_schema | mv_test |  5329 | 2020-02-14 02:22:26.328481 |
2020-02-14 02:22:28.369217 | Refresh successfully recomputed MV from scratch     |
Auto
dev     |    138 | mv_schema | mv_test |  5290 | 2020-02-14 02:21:56.885093 |
2020-02-14 02:21:56.885098 | Refresh failed. MV was not found                    |
Manual
```

SVL_QERROR

La visualizzazione SVL_QERROR è obsoleta.

SVL_QLOG

La visualizzazione SVL_QLOG contiene una log di tutte le query eseguite sul database.

Amazon Redshift crea la visualizzazione SVL_QLOG come sottoinsieme di informazioni leggibile dalla tabella STL_QUERY. Utilizzare questa tabella per trovare l'ID query di una query eseguita di recente o per vedere quanto tempo ha richiesto il completamento di una query.

SVL_QLOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. È possibile utilizzare questo ID per unire varie altre tabelle e visualizzazioni di sistema.

Nome colonna	Tipo di dati	Descrizione
xid	bigint	ID transazione.
pid	integer	ID di processo associato alla query.
starttime	timestamp	L'ora esatta in cui è iniziata l'esecuzione dell'istruzione, con 6 cifre di precisione per le frazioni di secondo, ad esempio: **`2009-06-12 11:29:19.131358`** .
endtime	timestamp	L'ora esatta in cui è terminata l'esecuzione dell'istruzione, con 6 cifre di precisione per le frazioni di secondo, ad esempio: **`2009-06-12 11:29:19.193640`** .
elapsed	bigint	Periodo di tempo in cui la query è stata eseguita (in microsecondi).
aborted	integer	Se la query è stata interrotta dal sistema o annullata dall'utente, questa colonna contiene **`1`**. Se la query è stata completata, questa colonna contiene **`0`**. Anche le query che vengono annullate per scopi di gestione del carico di lavoro e vengono successivamente riavviate hanno un valore di **`1`** in questa colonna.
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etiche tta definita con un comando SET QUERY GROUP. Se la query non è basata su file o non è impostato il parametro QUERY_GROUP, questo valore del campo è `default`.
substring	character(60)	Testo query troncato.
source_query	integer	Se la query ha utilizzato la cache dei risultati, l'ID query della query che è stata la fonte dei risultati memorizzati nella cache. Se la cache dei risultati non è stata utilizzata, questo valore di campo è `NULL`.

Nome colonna	Tipo di dati	Descrizione
concurrency_scaling_status_txt	text	Una descrizione dell'esecuzione della query sul cluster principale o sul cluster di dimensionamento della concorrenza.
from_sp_call	integer	Se la query è stata chiamata da una procedura archiviata, l'ID query della chiamata di procedura. Se la query non è stata eseguita come parte della procedura archiviata, il campo è NULL.

Query di esempio

L'esempio seguente restituisce l'ID query, l'orario di esecuzione e il testo query troncato delle cinque query di database più recenti eseguite dall'utente con `userid = 100`.

```
select query, pid, elapsed, substring from svl_qlog
where userid = 100
order by starttime desc
limit 5;

 query  |  pid  | elapsed  |           substring
--------+-------+----------+-----------------------------------------------
 187752 | 18921 | 18465685 | select query, elapsed, substring from svl_...
 204168 |  5117 |    59603 | insert into testtable values (100);
 187561 | 17046 |  1003052 | select * from pg_table_def where tablename...
 187549 | 17046 |  1108584 | select * from STV_WLM_SERVICE_CLASS_CONFIG
 187468 | 17046 |  5670661 | select * from pg_table_def where schemaname...
(5 rows)
```

L'esempio seguente restituisce il nome dello script SQL (colonna LABEL) e il tempo trascorso per una query che è stata annullata (**`aborted=1`**):

```
select query, elapsed, trim(label) querylabel
from svl_qlog where aborted=1;

 query | elapsed  |       querylabel
-------+----------+-------------------------
```

```
    16 |  6935292 | alltickittablesjoin.sql
(1 row)
```

SVL_QUERY_METRICS

La visualizzazione SVL_QUERY_METRICS mostra i parametri per le query completate. Questa visualizzazione viene derivata dalla tabella di sistema STL_QUERY_METRICS. Utilizza i valori in questa visualizzazione come aiuto per determinare i valori di soglia per la definizione delle regole di monitoraggio delle query. Per ulteriori informazioni, consulta Regole di monitoraggio delle query WLM.

SVL_QUERY_METRICS è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha eseguito la query che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
service_class	integer	ID della coda di query WLM (classe di servizio). Le code di query sono definite nella configurazione WLM. I parametri sono restituiti solo per le code definite dall'utente. Per un elenco degli ID delle classi di servizio, consultare ID classe di servizio WLM.
dimension	varchar(24)	Dimensione su cui è riportato il parametro. I valori possibili sono query, segmento, fase.
segment	integer	Numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. I segmenti di query

Nome colonna	Tipo di dati	Descrizione
		possono essere eseguiti in parallelo. Ogni segmento viene eseguito in un singolo processo. Se il valore del segmento è 0, viene eseguito il rollup dei valori di segmento dei parametri a livello della query.
step	integer	ID per il tipo di fase che è stata eseguita. La descrizione del tipo di fase viene mostrata nella colonna `step_label ..`
step_label	varchar(30)	Tipo di fase che è stata eseguita.
query_cpu_time	bigint	Tempo di CPU utilizzato dalla query, in secondi. Il tempo di CPU è diverso dal tempo di esecuzione della query.
query_blocks_read	bigint	Numero di blocchi da 1 MB letti dalla query.
query_execution_time	bigint	Tempo di esecuzione trascorso per una query, in secondi. Il tempo di esecuzione non include il tempo trascorso in attesa in una coda. Vedi `query_queue_time` per il tempo di permanenza in coda.
query_cpu_usage_percent	bigint	Percentuale di capacità della CPU utilizzata dalla query.
query_temp_blocks_to_disk	bigint	La quantità di spazio su disco utilizzata da una query per scrivere risultati intermedi, in MB.
segment_execution_time	bigint	Tempo di esecuzione trascorso per un singolo segmento, in secondi.
cpu_skew	numeric(38,2)	Il rapporto tra l'utilizzo di CPU massimo per ogni sezione e l'utilizzo di CPU medio per tutte le sezioni. Questo parametro viene definito a livello di segmento.
io_skew	numeric(38,2)	Il rapporto tra i blocchi massimi letti (I/O) per ogni sezione di blocchi medi letti per tutte le sezioni.

Nome colonna	Tipo di dati	Descrizione
scan_row_count	bigint	Il numero di righe in una fase di scansione. Il conteggio delle righe è il numero totale di righe generate prima di aver applicato filtri alle righe contrassegnate per l'eliminazione (righe fantasma) e prima dell'applicazione di filtri di query definiti dall'utente.
join_row_count	bigint	Il numero di righe elaborate in una fase di unione.
nested_loop_join_row_count	bigint	Il numero di righe di un join annidiato dei cicli.
return_row_count	bigint	Il numero di righe restituite dalla query.
spectrum_scan_row_count	bigint	Il numero di righe di cui è stata eseguita la scansione da Amazon Redshift Spectrum in Amazon S3.
spectrum_scan_size_mb	bigint	La quantità di dati, in MB, di cui è stata eseguita la scansione da Amazon Redshift Spectrum in Amazon S3.
query_queue_time	bigint	La quantità di tempo espressa in secondi di permanenza della query nella coda.

SVL_QUERY_METRICS_SUMMARY

La visualizzazione SVL_QUERY_METRICS_SUMMARY mostra i valori massimi dei parametri per le query completate. Questa visualizzazione viene derivata dalla tabella di sistema STL_QUERY_METRICS. Utilizza i valori in questa visualizzazione come aiuto per determinare i valori di soglia per la definizione delle regole di monitoraggio delle query. Per ulteriori informazioni su regole e metriche per il monitoraggio delle query per Amazon Redshift, consulta Regole di monitoraggio delle query WLM.

SVL_QUERY_METRICS_SUMMARY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha eseguito la query che ha generato la voce.
query	integer	ID query. La colonna di query può essere utilizzata per unire altre tabelle e visualizzazioni del sistema.
service_class	integer	ID della coda di query WLM (classe di servizio). Le code di query sono definite nella configurazione WLM. I parametri sono restituiti solo per le code definite dall'utente. Per un elenco degli ID delle classi di servizio, consulta ID classe di servizio WLM.
query_cpu_time	bigint	Tempo di CPU utilizzato dalla query, in secondi. Il tempo di CPU è diverso dal tempo di esecuzione della query.
query_blocks_read	bigint	Numero di blocchi da 1 MB letti dalla query.
query_execution_time	bigint	Tempo di esecuzione trascorso per una query, in secondi. Il tempo di esecuzione non include il tempo trascorso in attesa in una coda.
query_cpu_usage_percent	numeric(38,2)	Percentuale di capacità della CPU utilizzata dalla query.
query_temp_blocks_to_disk	bigint	La quantità di spazio su disco utilizzata da una query per scrivere risultati intermedi, in MB.
segment_execution_time	bigint	Tempo di esecuzione trascorso per un singolo segmento, in secondi.

Nome colonna	Tipo di dati	Descrizione
cpu_skew	numeric(38,2)	Il rapporto tra l'utilizzo di CPU massimo per ogni sezione e l'utilizzo di CPU medio per tutte le sezioni. Questo parametro viene definito a livello di segmento.
io_skew	numeric(38,2)	Il rapporto tra i blocchi massimi letti (I/O) per ogni sezione di blocchi medi letti per tutte le sezioni.
scan_row_count	bigint	Il numero di righe in una fase di scansione. Il conteggio delle righe è il numero totale di righe generate prima di aver applicato filtri alle righe contrassegnate per l'eliminazione (righe fantasma) e prima dell'applicazione di filtri di query definiti dall'utente.
join_row_count	bigint	Il numero di righe elaborate in una fase di unione.
nested_loop_join_row_count	bigint	Il numero di righe di un join annidiato dei cicli.
return_row_count	bigint	Il numero di righe restituite dalla query.
spectrum_scan_row_count	bigint	Il numero di righe di cui è stata eseguita la scansione da Amazon Redshift Spectrum in Amazon S3.
spectrum_scan_size_mb	bigint	La quantità di dati, in MB, di cui è stata eseguita la scansione da Amazon Redshift Spectrum in Amazon S3.
query_queue_time	bigint	La quantità di tempo espressa in secondi di permanenza della query nella coda.

SVL_QUERY_QUEUE_INFO

Riepiloga i dettagli delle query che hanno impiegato tempo in una coda di query di gestione del carico di lavoro (WLM) o in una coda di commit.

La visualizzazione SVL_QUERY_QUEUE_INFO filtra le query eseguite dal sistema e mostra solo le query eseguite da un utente.

La visualizzazione SVL_QUERY_QUEUE_INFO riepiloga le informazioni dalla STL_QUERY, STL_WLM_QUERY e STL_COMMIT_STATS dalle tabelle di sistema.

SVL_QUERY_QUEUE_INFO è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
database	text	Il nome del database al quale l'utente era collegato al momento del rilascio della query.
query	integer	ID query.
xid	bigint	ID transazione.
userid	integer	ID dell'utente che ha generato la query.
querytxt	text	Primi 100 caratteri del testo della query.
queue_start_time	timestamp	Orario in UTC in cui la query è stata inserita nella coda WLM.
exec_start_time	timestamp	Orario in UTC in cui è stata avviata l'esecuzione della query.
service_class	integer	ID per la classe di servizio. Le classi di servizio vengono definite nel file di configurazione WLM.
slots	integer	Numero di slot di query WLM.
queue_elapsed	bigint	Tempo trascorso dalla query in attesa nella coda WLM (in secondi).
exec_elapsed	bigint	Tempo impiegato per l'esecuzione della query (in secondi).

Nome colonna	Tipo di dati	Descrizione
wlm_total_elapsed	bigint	Tempo trascorso dalla query in una coda WLM (queue_elapsed), più il tempo impiegato per l'esecuzione della query (exec_elapsed).
commit_queue_elapsed	bigint	Tempo trascorso dalla query in attesa nella coda commit (in secondi).
commit_exec_time	bigint	Tempo trascorso dalla query nell'operazione commit (in secondi).
service_class_name	character(64)	Il nome della classe dei servizi.

Query di esempio

L'esempio seguente mostra il tempo trascorso dalle query nelle code WLM.

```
select query, service_class, queue_elapsed, exec_elapsed, wlm_total_elapsed
from svl_query_queue_info
where wlm_total_elapsed > 0;

  query  | service_class | queue_elapsed | exec_elapsed | wlm_total_elapsed
---------+---------------+---------------+--------------+-------------------
 2742669 |             6 |             2 |          916 |               918
 2742668 |             6 |             4 |          197 |               201
(2 rows)
```

SVL_QUERY_REPORT

Amazon Redshift crea la visualizzazione SVL_QUERY_REPORT da un'UNIONE di una serie di tabelle di sistema STL di Amazon Redshift per fornire informazioni sulle fasi delle query completate.

Questa vista suddivide le informazioni sulle query completate per sezione e per fase, che possono aiutare a risolvere i problemi relativi a nodi e sezioni nel cluster Amazon Redshift.

SVL_QUERY_REPORT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizzazioni di sistema.
sezione	integer	Sezione dei dati dove è stata eseguita la fase.
segment	integer	Numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. I segmenti di query possono essere eseguiti in parallelo. Ogni segmento viene eseguito in un singolo processo.
step	integer	FASI di query completate.
start_time	timestamp	Orario esatto in UTC in cui il segmento ha avviato l'esecuzione, con 6 cifre di precisione per le frazioni di secondo. Ad esempio: **`2012-12-12 11:29:19.131358`**
end_time	timestamp	Orario esatto in UTC in cui il segmento ha terminato l'esecuzione, con 6 cifre di precisione per le frazioni di secondo. Ad esempio: **`2012-12-12 11:29:19.131467`**
elapsed_time	bigint	Tempo (in microsecondi) in cui il segmento è stato eseguito.

Nome colonna	Tipo di dati	Descrizione
righe	bigint	Numero di righe prodotte dalla fase (per sezione). Questo numero rappresenta il numero di righe per la sezione che risultano dall'esec uzione della fase, non il numero di righe ricevute o elaborate dalla fase. In altre parole, questo è il numero di righe che superano la fase e passano a quella successiva.
bytes	bigint	Numero di byte prodotti dalla fase (per sezione).
etichetta	char(256)	Etichetta di fase, che consiste in un nome di fase di query e, quando applicabile, in un ID di tabella e in un nome di tabella (per esempio, `scan tbl=100448 name =user`). Gli ID di tabella a tre cifre fanno in genere riferimento alle scansioni delle tabelle transitorie. Quando viene visualizzato `tbl=0`, in genere fa riferimento a una scansione di un valore costante.
is_diskba sed	character (1)	Se questa fase della query è stata eseguita come operazione basata su disco: true (**`t`**) o false (**`f`**). Solo determinate fasi, come hash, sort e le fasi di aggregazione, possono accedere al disco. Molti tipi di fase sono sempre eseguiti in memoria.
workmem	bigint	Quantità di memoria di lavoro (in byte) assegnata alla fase di query. Questo valore è la soglia query_working_mem allocata per l'utilizzo durante l'esecuzione, non la quantità di memoria che è stata effettiva mente utilizzata
is_rrscan	character (1)	Se true (**`t`**), indica che in questa fase è stata utilizzata la scansione a intervallo limitato.
is_delaye d_scan	character (1)	Se true (**`t`**), indica che in questa fase è stata utilizzata la scansione ritardata.
rows_pre_ filter	bigint	Per le scansioni di tabelle permanenti, il numero totale di righe emesse prima di aver applicato filtri alle righe contrassegnate per l'elimina zione (righe fantasma) e prima dell'applicazione di filtri di query definiti dall'utente.

Query di esempio

La query seguente mostra la differenza di dati delle righe restituite per la query con l'ID query 279. Utilizzare questa query per determinare se i dati del database sono distribuiti in maniera uniforme sulle sezioni del cluster del data warehouse:

```
select query, segment, step, max(rows), min(rows),
case when sum(rows) > 0
then ((cast(max(rows) -min(rows) as float)*count(rows))/sum(rows))
else 0 end
from svl_query_report
where query = 279
group by query, segment, step
order by segment, step;
```

Questa query deve restituire dati simili al seguente output di esempio:

```
query | segment | step |   max    |   min    |         case
------+---------+------+----------+----------+----------------------
279 |       0 |    0 | 19721687 | 19721687 |                    0
279 |       0 |    1 | 19721687 | 19721687 |                    0
279 |       1 |    0 |   986085 |   986084 | 1.01411202804304e-06
279 |       1 |    1 |   986085 |   986084 | 1.01411202804304e-06
279 |       1 |    4 |   986085 |   986084 | 1.01411202804304e-06
279 |       2 |    0 |  1775517 |   788460 |     1.00098637606408
279 |       2 |    2 |  1775517 |   788460 |     1.00098637606408
279 |       3 |    0 |  1775517 |   788460 |     1.00098637606408
279 |       3 |    2 |  1775517 |   788460 |     1.00098637606408
279 |       3 |    3 |  1775517 |   788460 |     1.00098637606408
279 |       4 |    0 |  1775517 |   788460 |     1.00098637606408
279 |       4 |    1 |  1775517 |   788460 |     1.00098637606408
279 |       4 |    2 |        1 |        1 |                    0
279 |       5 |    0 |        1 |        1 |                    0
279 |       5 |    1 |        1 |        1 |                    0
279 |       6 |    0 |       20 |       20 |                    0
279 |       6 |    1 |        1 |        1 |                    0
279 |       7 |    0 |        1 |        1 |                    0
279 |       7 |    1 |        0 |        0 |                    0
(19 rows)
```

SVL_QUERY_SUMMARY

Utilizzare la visualizzazione SVL_QUERY_SUMMARY per trovare informazioni generali riguardanti l'esecuzione di una query.

La visualizzazione SVL_QUERY_SUMMARY contiene un sottoinsieme di dati della visualizzazione SVL_QUERY_REPORT. Si noti che le informazioni in SVL_QUERY_SUMMARY sono aggregate da tutti i nodi.

> **Note**
>
> La vista SVL_QUERY_SUMMARY contiene solo informazioni sulle query eseguite da Amazon Redshift, non su altri comandi di utilità e DDL. Per le informazioni complete e un elenco di tutte le istruzioni eseguite da Amazon Redshift, compresi i comandi di utilità e DDL, è possibile anche eseguire una query sulla vista SVL_STATEMENTTEXT.

SVL_QUERY_SUMMARY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Per informazioni su SVCS_QUERY_SUMMART, consultare SVCS_QUERY_SUMMARY.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
query	integer	ID query. Consente di unire in join varie altre tabelle e visualizz azioni di sistema.
stm	integer	Flusso: un insieme di segmenti simultanei in una query. Una query ha uno o più flussi.

Nome colonna	Tipo di dati	Descrizione
seg	integer	Numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. I segmenti di query possono essere eseguiti in parallelo. Ogni segmento viene eseguito in un singolo processo.
step	integer	La fase di query eseguita.
maxtime	bigint	Quantità di tempo massima per l'esecuzione della fase (in microsecondi).
avgtime	bigint	Tempo medio per l'esecuzione della fase (in microsecondi).
righe	bigint	Numero di righe di dati coinvolte nella fase di query.
byte	bigint	Numero di byte di dati coinvolti nella fase di query.
rate_row	double precision	Velocità di esecuzione di query per riga.
rate_byte	double precision	Velocità di esecuzione di query per byte.
etichetta	text	Etichetta di fase, che consiste in un nome di fase di query e, quando applicabile, in un ID di tabella e in un nome di tabella (per esempio, scan tbl=100448 name =user). Gli ID di tabella a tre cifre fanno in genere riferimento alle scansioni delle tabelle transitorie. Quando viene visualizzato `tbl=0`, in genere fa riferimento a una scansione di un valore costante.
is_diskbased	character(1)	Se questa fase della query è stata eseguita come operazione basata su disco su qualsiasi nodo nel cluster: true (**`t`**) o false (**`f`**). Solo determinate fasi, come hash, sort e le fasi di aggregazione, possono accedere al disco. Molti tipi di fase sono sempre eseguiti in memoria.
workmem	bigint	Quantità di memoria di lavoro (in byte) assegnata alla fase di query.

Nome colonna	Tipo di dati	Descrizione
is_rrscan	character(1)	Se true (**t**), indica che in questa fase è stata utilizzata la scansione a intervallo limitato. Il valore predefinito è false (**f**).
is_delayed_scan	character(1)	Se true (**t**), indica che in questa fase è stata utilizzata la scansione ritardata. Il valore predefinito è false (**f**).
rows_pre_filter	bigint	Per le scansioni di tabelle permanenti, il numero totale di righe emesse prima di aver applicato filtri alle righe contrassegnate per l'eliminazione (righe fantasma).

Query di esempio

Visualizzazione delle informazioni di elaborazione per una fase di query

La query seguente mostra le informazioni di elaborazione di base per ogni fase della query 87:

```
select query, stm, seg, step, rows, bytes
from svl_query_summary
where query = 87
order by query, seg, step;
```

Questa query recupera le informazioni di elaborazione relative alla query 87, come mostrato nel seguente output di esempio:

```
 query | stm | seg | step |  rows  |  bytes
-------+-----+-----+------+--------+---------
87     |   0 |   0 |    0 |     90 |    1890
87     |   0 |   0 |    2 |     90 |     360
87     |   0 |   1 |    0 |     90 |     360
87     |   0 |   1 |    2 |     90 |    1440
87     |   1 |   2 |    0 | 210494 | 4209880
87     |   1 |   2 |    3 |  89500 |       0
87     |   1 |   2 |    6 |      4 |      96
87     |   2 |   3 |    0 |      4 |      96
87     |   2 |   3 |    1 |      4 |      96
87     |   2 |   4 |    0 |      4 |      96
87     |   2 |   4 |    1 |      1 |      24
```

```
87     |   3 |   5 |    0 |      1 |      24
87     |   3 |   5 |    4 |      0 |       0
(13 rows)
```

Determinare se le fasi della query si sono riversate sul disco

La query seguente mostra se una delle fasi della query con l'ID query 1025 si è riversata o meno sul disco (consultare la visualizzazione SVL_QLOG per scoprire come ottenere l'ID query di una query) o se la query è stata eseguita interamente in memoria:

```
select query, step, rows, workmem, label, is_diskbased
from svl_query_summary
where query = 1025
order by workmem desc;
```

Questa query restituisce il seguente output di esempio:

```
query| step|  rows  |  workmem   |  label        | is_diskbased
-----+-----+--------+-----------+---------------+--------------
1025 |  0  |16000000|  141557760 |scan tbl=9     | f
1025 |  2  |16000000|  135266304 |hash tbl=142   | t
1025 |  0  |16000000|  128974848 |scan tbl=116536| f
1025 |  2  |16000000|  122683392 |dist           | f
(4 rows)
```

Attraverso la scansione dei valori di IS_DISKBASED, è possibile vedere quali fasi della query sono andate sul disco. Per la query 1025, la fase di hash è stata eseguita su disco. Le fasi che possono essere eseguite su disco includono hash, aggr e le fasi di ordinamento. Per visualizzare solo le fasi di query basate su disco, aggiungere la clausola **`and is_diskbased = 't'`** all'istruzione SQL nell'esempio precedente.

SVL_RESTORE_ALTER_TABLE_PROGRESS

Utilizzate SVL_RESTORE_ALTER_TABLE_PROGRESS per monitorare l'avanzamento della migrazione di ogni tabella nel cluster durante un ridimensionamento classico ai nodi RA3. Acquisisce la velocità di trasmissione effettiva storica della migrazione dei dati durante l'operazione di ridimensionamento. Per ulteriori informazioni sul ridimensionamento classico ai nodi RA3, vai a Ridimensionamento classico.

SVL_RESTORE_ALTER_TABLE_PROGRESS è visibile solo ai superutenti. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_RESTORE_LOG. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

> **Note**
>
> Le righe con un avanzamento di o vengono eliminate dopo 7 giorni. `100.00%` `ABORTED` Le righe delle tabelle eliminate durante o dopo un ridimensionamento classico possono ancora apparire in SVL_RESTORE_ALTER_TABLE_PROGRESS.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
tbl	integer	L'ID della tabella.
progress	char(32)	Lo stato dell'avanzamento della ridistribuzione della tabella. I valori `0.00%` possibili `100.00%` sono `ABORTED` le percentuali comprese tra a e il messaggio. `ABORTED`significa che la ridistribuzione è stata interrotta senza terminare, con il motivo spiegato nella `message` colonna.
message	char(256)	Il messaggio associato all'avanzamento della ridistribuzione della tabella.

Query di esempio

La seguente query restituisce query in esecuzione e in coda.

```
select * from svl_restore_alter_table_progress;

tbl     | progress |                          message
--------+----------+-------------------------------------------------------------
105614  | ABORTED  | Abort:Table no longer contains the prior dist key column.
105610  | ABORTED  | Abort:Table no longer contains the prior dist key column.
105594  | 0.00%    | Table waiting for alter diststyle conversion.
105602  | ABORTED  | Abort:Table no longer contains the prior dist key column.
```

```
105606  | ABORTED  | Abort:Table no longer contains the prior dist key column.
105598  | 100.00%  | Restored to distkey successfully.
```

SVL_S3LIST

Utilizza la vista SVL_S3LIST per ottenere dettagli sulle query Amazon Redshift Spectrum a livello di segmento.

SVL_S3LIST è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> SVL_S3LIST contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_EXTERNAL_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query	integer	L'ID di query.
segment	integer	Il numero di segmento. Una query è costituito da più segmenti.
nodo	integer	Il numero di nodo.
sezione	integer	La sezione di dati su cui un particolare segmento è stato eseguito.
eventtime	timestamp	L'orario in formato UTC in cui l'evento viene registrato.
bucket	text	Nome del bucket Amazon S3.

Nome colonna	Tipo di dati	Descrizione
prefisso	text	Il prefisso del posizione del bucket Amazon S3.
recursive	char(1)	Qualora ci sia la scansione ricorsiva delle sottocartelle.
retrieved_files	integer	Il numero di file elencati.
max_file_size	bigint	La dimensione massima dei file tra i file elencati.
vg_file_size	double precision	La dimensione media dei file tra i file elencati.
generated_splits	integer	Il numero di suddivisioni dei file.
avg_split_length	double precision	La lunghezza media delle suddivisioni dei file in byte.
durata	bigint	La durata dell'elenco dei file in microsecondi.

Query di esempio

Le query SVL_S3LIST di esempio seguenti per l'ultima query da eseguire.

```
select *
from svl_s3list
where query = pg_last_query_id()
order by query,segment;
```

SVL_S3LOG

Utilizza la vista SVL_S3LOG per ottenere dettagli riguardanti le query Amazon Redshift Spectrum a livello di segmento e di sezione di nodo.

SVL_S3LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> SVL_S3LOG contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_EXTERNAL_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
pid	integer	L'ID di processo.
query	integer	L'ID di query.
segment	integer	Il numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi.
step	integer	La fase di query eseguita.
nodo	integer	Il numero di nodo.
sezione	integer	La sezione di dati su cui un particolare segmento è stato eseguito.
eventtime	timestamp	Orario in UTC in cui la fase ha iniziato l'esecuzione.
message	text	Messaggio per la voce di log.

Query di esempio

Le query SVCS_S3LOG di esempio seguenti per l'ultima query eseguita.

```
select *
from svl_s3log
where query = pg_last_query_id()
order by query,segment,slice;
```

SVL_S3PARTITION

Utilizza la vista SVL_S3PARTITION per ottenere dettagli riguardanti le partizioni Amazon Redshift Spectrum a livello di segmento e di sezione di nodo.

SVL_S3PARTITITION è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> SVL_S3PARTITION contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_EXTERNAL_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query	integer	L'ID di query.
segment	integer	Un numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi.
nodo	integer	Il numero di nodo.
sezione	integer	La sezione di dati su cui un particolare segmento è stato eseguito.

Nome colonna	Tipo di dati	Descrizione
starttime	timestamp without time zone	Orario in UTC in cui l'eliminazione delle partizioni ha iniziato l'esecuzione.
endtime	timestamp without time zone	Orario in UTC in cui l'eliminazione delle partizioni è stata completata.
durata	bigint	Tempo trascorso (in microsecondi).
total_partitions	integer	Numero di partizioni totali.
qualified_partitions	integer	Numero di partizioni qualificate.
assigned_partitions	integer	Numero di partizioni assegnate sulla sezione.
incarico	carattere	Tipo di incarico.

Query di esempio

L'esempio seguente dà i dettagli di partizione per l'ultima query completata.

```
SELECT query, segment,
       MIN(starttime) AS starttime,
       MAX(endtime) AS endtime,
       datediff(ms,MIN(starttime),MAX(endtime)) AS dur_ms,
       MAX(total_partitions) AS total_partitions,
       MAX(qualified_partitions) AS qualified_partitions,
       MAX(assignment) as assignment_type
FROM svl_s3partition
WHERE query=pg_last_query_id()
GROUP BY query, segment
```

```
query | segment |           starttime           |           endtime           | dur_ms|
 total_partitions | qualified_partitions | assignment_type
```

```
------+---------+-------------------------------+----------------------------+-------
+-----------------+---------------------+---------------
99232 |       0 | 2018-04-17 22:43:50.201515    | 2018-04-17 22:43:54.674595  |  4473 |
       2526       |        334           | p
```

SVL_S3PARTITION_SUMMARY

Utilizza la vista SVL_S3PARTITION_SUMMARY per ottenere un riepilogo dell'elaborazione delle partizioni delle query Redshift Spectrum a livello di segmento.

SVL_S3PARTITION_SUMMARY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Per informazioni su SVCS_S3PARTITION, consultare SVCS_S3PARTITION_SUMMARY.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query	integer	L'ID di query. È possibile utilizzare questo valore per unire varie altre tabelle e visualizzazioni di sistema.
segment	integer	Il numero di segmento. Una query è costituito da più segmenti.
incarico	char(1)	Il tipo di incarico di partizione tra nodi.
min_start time	timestamp	L'orario in formato UTC in cui l'elaborazione delle partizion i è iniziata.
max_endti me	timestamp	L'orario in formato UTC in cui l'elaborazione delle partizioni è stata completata.
min_durat ion	bigint	Il periodo minimo di elaborazione delle partizioni utilizzato da un nodo per questa query (in microsecondi).
max_durat ion	bigint	Il periodo massimo di elaborazione delle partizioni utilizzato da un nodo per questa query (in microsecondi).

Nome colonna	Tipo di dati	Descrizione
avg_duration	bigint	Il periodo medio di elaborazione delle partizioni utilizzato da un nodo per questa query (in microsecondi).
total_partitions	integer	Il numero totale di partizioni in una tabella esterna.
qualified_partitions	integer	Il numero totale di partizioni qualificate.
min_assigned_partitions	integer	Il numero minimo di partizioni assegnate su un nodo.
min_assigned_partitions	integer	Il numero massimo di partizioni assegnate su un nodo.
Il numero minimo di partizioni assegnate su un nodo.	bigint	Il numero medio di partizioni assegnate su un nodo.

Query di esempio

L'esempio seguente dà i dettagli di partizione per l'ultima query completata.

```
select query, segment, assignment, min_starttime, max_endtime, min_duration,
 avg_duration
from svl_s3partition_summary
where query = pg_last_query_id()
order by query,segment;
```

SVL_S3QUERY

Utilizza la vista SVL_S3QUERY per ottenere dettagli riguardanti le query Amazon Redshift Spectrum a livello di segmento e di sezione di nodo.

SVL_S3QUERY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

> **Note**
>
> SVL_S3QUERY contiene solo le query eseguite sui cluster principali. Non contiene query eseguite su cluster con dimensionamento simultaneo. Per accedere alle query eseguite sui cluster principali e con dimensionamento simultaneo, ti consigliamo di utilizzare la vista di monitoraggio SYS SYS_EXTERNAL_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato una certa voce.
query	integer	L'ID di query.
segment	integer	Un numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi.
step	integer	La fase di query eseguita.
nodo	integer	Il numero di nodo.
sezione	integer	La sezione di dati su cui un particolare segmento è stato eseguito.
starttime	timestamp	Orario in UTC in cui la query ha iniziato l'esecuzione.

Nome colonna	Tipo di dati	Descrizione
endtime	timestamp	Orario in UTC in cui è stata completata l'esecuzione della query
elapsed	integer	Tempo trascorso (in microsecondi).
external_table_name	char(136)	Formato interno del nome tabella esterno per la fase di scansione s3.
is_partitioned	char(1)	Se true (**t**), questo valore della colonna indica che la tabella esterna è partizionata.
is_rrscan	char(1)	Se true (**t**), questo valore della colonna indica che è stata applicata una scansione a intervallo limitato.
s3_scanned_rows	bigint	Il numero di righe di cui è stata eseguita la scansione da Amazon S3 e che sono state inviate al livello Redshift Spectrum.
s3_scanned_bytes	bigint	Il numero di byte di cui è stata eseguita la scansione da Amazon S3 e che sono stati inviati al livello Redshift Spectrum.
s3query_returned_rows	bigint	Il numero di righe restituite dal livello Redshift Spectrum al cluster.
s3query_returned_bytes	bigint	Il numero di byte restituiti dal livello Redshift Spectrum al cluster.
files	integer	Il numero di file elaborati per questa fase di scansione S3 in questa sezione.

Nome colonna	Tipo di dati	Descrizione
splits	int	Il numero di suddivisioni elaborate in questa sezione. Con file di dati di grandi dimensioni che possono essere suddivisi, per esempio file di dimensioni superiori a circa 512 MB, Redshift Spectrum cerca di dividere i file in più richieste S3 per l'elaborazione parallela.
total_split_size	bigint	La dimensione totale di tutte le suddivisioni elaborate in questa sezione, in byte.
max_split_size	bigint	La dimensione massima di suddivisione elaborata per questa sezione, in byte.
total_retries	integer	Il numero totale di tentativi per i file elaborati.
max_retries	integer	Il numero massimo di tentativi per un singolo file elaborato.
max_request_duration	integer	La durata massima di una singola richiesta Redshift Spectrum (in microsecondi).
avg_request_duration	double precision	La durata media delle richieste Redshift Spectrum (in microsecondi).
max_request_parallelism	integer	Il numero massimo di Redshift Spectrum in sospeso su questa sezione per questa fase di scansione S3.
avg_request_parallelism	double precision	Il numero medio di richieste parallele Redshift Spectrum su questa sezione per questa fase di scansione S3.

Query di esempio

L'esempio seguente dà i dettagli della fase di scansione per l'ultima query completata.

```
select query, segment, slice, elapsed, s3_scanned_rows, s3_scanned_bytes,
 s3query_returned_rows, s3query_returned_bytes, files
from svl_s3query
where query = pg_last_query_id()
order by query,segment,slice;
```

```
query | segment | slice | elapsed | s3_scanned_rows | s3_scanned_bytes |
 s3query_returned_rows | s3query_returned_bytes | files
------+---------+-------+---------+-----------------+------------------
+----------------------+------------------------+------
 4587 |       2 |     0 |   67811 |               0 |                0 |
      0 |                      0 |     0
 4587 |       2 |     1 |  591568 |          172462 |         11260097 |
   8513 |                  170260 |     1
 4587 |       2 |     2 |  216849 |               0 |                0 |
      0 |                      0 |     0
 4587 |       2 |     3 |  216671 |               0 |                0 |
      0 |                      0 |     0
```

SVL_S3QUERY_SUMMARY

Utilizza la vista SVL_S3QUERY_SUMMARY per ottenere un riepilogo di tutte le query Amazon Redshift Spectrum (query S3) che sono state eseguite nel sistema. SVL_S3QUERY_SUMMARY aggrega i dettagli da SVL_S3QUERY a livello di segmento.

SVL_S3QUERY_SUMMARY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_EXTERNAL_QUERY_DETAIL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Per SVCS_S3QUERY_SUMMART, consultare SVCS_S3QUERY_SUMMARY.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato quella determinata voce.
query	integer	L'ID di query. È possibile utilizzare questo valore per unire varie altre tabelle e visualizzazioni di sistema.
xid	bigint	L'ID transazione.
pid	integer	L'ID di processo.
segment	integer	Il numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi.
step	integer	La fase di query eseguita.
starttime	timestamp	Orario in UTC in cui la query ha iniziato l'esecuzione.
endtime	timestamp	Orario in UTC in cui è stata completata la query.
elapsed	integer	Il periodo di tempo in cui la query è stata eseguita (in microsecondi).
aborted	integer	Se la query è stata interrotta dal sistema o annullata dall'utente, questa colonna contiene **1**. Se la query è stata completata, questa colonna contiene **0**.
external_table_name	char(136)	Il formato interno del nome esterno della tabella per la scansione della tabella esterna.
file_format	character(16)	Il formato del file dei dati della tabella esterna.
is_partitioned	char(1)	Se true (**t**), questo valore della colonna indica che la tabella esterna è partizionata.

Nome colonna	Tipo di dati	Descrizione
is_rrscan	char(1)	Se true (**t**), questo valore della colonna indica che è stata applicata una scansione a intervallo limitato.
is_nested	char(1)	Se true (**t**), questo valore della colonna indica che viene eseguito l'accesso al tipo di dati della colonna nidificata.
s3_scanned_rows	bigint	Il numero di righe di cui è stata eseguita la scansione da Amazon S3 e che sono state inviate al livello Redshift Spectrum.
s3_scanned_bytes	bigint	Il numero di byte di cui è stata eseguita la scansione da Amazon S3 e che sono stati inviati al livello Redshift Spectrum sulla base di dati compressi.
s3query_returned_rows	bigint	Il numero di righe restituite dal livello Redshift Spectrum al cluster.
s3query_returned_bytes	bigint	Il numero di byte restituiti dal livello Redshift Spectrum al cluster. Una grande quantità di dati restituiti a Amazon Redshift può influire sulle prestazioni di sistema.
files	integer	Il numero di file elaborati per questa query Redshift Spectrum. Un piccolo numero di file limita i vantaggi dell'elaborazione parallela.
files_max	integer	Il numero massimo di file elaborati in una sezione.
files_avg	integer	Il numero medio di file elaborati in una sezione.
splits	int	Il numero di suddivisioni elaborate per questo segmento. Il numero di suddivisioni elaborate in questa sezione. Con file di dati di grandi dimensioni che possono essere suddivisi, per esempio file di dimensioni superiori a circa 512 MB, Redshift Spectrum cerca di dividere i file in più richieste S3 per l'elaborazione parallela.

Nome colonna	Tipo di dati	Descrizione
splits_max	int	Il numero massimo di suddivisioni elaborate in questa sezione.
splits_avg	int	Il numero medio di suddivisioni elaborate in questa sezione.
total_split_size	bigint	La dimensione totale di tutte le suddivisioni elaborate.
max_split_size	bigint	La dimensione massima di suddivisione elaborata, in byte.
avg_split_size	bigint	La dimensione media di suddivisione elaborata, in byte.
total_retries	integer	Il numero totale di tentativi per un singolo file elaborato.
max_retries	integer	Il numero massimo di tentativi per qualsiasi dei file elaborati.
max_request_duration	integer	La durata massima di una singola richiesta di file (in microsecondi). Le query con un'esecuzione lunga possono indicare un collo di bottiglia.
avg_request_duration	double precision	La durata media delle richieste di file (in microsecondi).
max_request_parallelism	integer	Il numero massimo di richieste parallele in una sezione per questa query Redshift Spectrum.
avg_request_parallelism	double precision	Il numero medio di richieste parallele in una sezione per questa query Redshift Spectrum.

Nome colonna	Tipo di dati	Descrizione
total_slowdown_count	bigint	Il numero totale di richieste Amazon S3 con un errore di rallentamento verificatosi durante la scansione della tabella esterna.
max_slowdown_count	integer	Il numero massimo di richieste di Amazon S3 con un errore di rallentamento verificatosi durante la scansione della tabella esterna in una sezione.

Query di esempio

L'esempio seguente dà i dettagli della fase di scansione per l'ultima query completata.

```
select query, segment, elapsed, s3_scanned_rows, s3_scanned_bytes,
 s3query_returned_rows, s3query_returned_bytes, files
from svl_s3query_summary
where query = pg_last_query_id()
order by query,segment;
```

```
query | segment | elapsed | s3_scanned_rows | s3_scanned_bytes | s3query_returned_rows
 | s3query_returned_bytes | files
------+---------+---------+-----------------+------------------+----------------------
+------------------------+------
 4587 |       2 |   67811 |               0 |                0 |                    0
 |                      0 |     0
 4587 |       2 |  591568 |          172462 |         11260097 |                 8513
 |                 170260 |     1
 4587 |       2 |  216849 |               0 |                0 |                    0
 |                      0 |     0
 4587 |       2 |  216671 |               0 |                0 |                    0
 |                      0 |     0
```

SVL_S3RETRIES

Utilizza la vista SVL_S3RETRIES per ottenere informazioni sul perché una query Amazon Redshift Spectrum basata su Amazon S3 non è andata a buon fine.

SVL_S3RETRIES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione		
query	integer	L'ID di query.		
segment	integer	Numero di segmento. Una query consiste in più segmenti e ogni segmento consiste in una o più fasi. I segmenti di query possono essere eseguiti in parallelo. Ogni segmento viene eseguito in un singolo processo.		
nodo	integer	Il numero di nodo.		
sezione	integer	La sezione di dati su cui un particolare segmento è stato eseguito.		
eventtime	timestamp without time zone	Orario in UTC in cui la fase ha iniziato l'esecuzione.		
retries	integer	Il numero di tentativi per la query.		
successful_fetches	integer	Il numero di volte in cui i dati sono stati restituiti.		

Nome colonna	Tipo di dati	Descrizione		
file_size	bigint	Questa dimensioni del file in byte.		
posizione	text	La posizione della tabella.		
message	text	Messaggio di errore.		

Query di esempio

L'esempio seguente recupera i dati sulle query S3 che non hanno avuto esito positivo.

```
SELECT svl_s3retries.query, svl_s3retries.segment, svl_s3retries.node,
 svl_s3retries.slice, svl_s3retries.eventtime, svl_s3retries.retries,
svl_s3retries.successful_fetches, svl_s3retries.file_size,
 btrim((svl_s3retries."location")::text) AS "location",
 btrim((svl_s3retries.message)::text)
AS message FROM svl_s3retries;
```

SVL_SPATIAL_SIMPLIFY

Puoi eseguire una query nella vista di sistema SVL_SPATIAL_SIMPLIFY per ottenere informazioni sugli oggetti della geometria spaziale semplificata utilizzando il comando COPY. Quando utilizzi COPY in uno shapefile, puoi specificare le opzioni di importazione `tolerance`, SIMPLIFY AUTO, and SIMPLIFY AUTO `max_tolerance`. Il risultato della semplificazione è riepilogato nella vista di sistema SVL_SPATIAL_SIMPLIFY.

Quando si imposta SIMPLIFY AUTO `max_tolerance`, questa vista contiene una riga per ogni geometria che ha superato la dimensione massima. Quando è impostato SIMPLIFY `tolerance`, viene memorizzata una riga per l'intera operazione COPY. Questa riga fa riferimento all'ID query COPY e alla tolleranza di semplificazione specificata.

SVL_SPATIAL_SIMPLIFY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_SPATIAL_SIMPLIFY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query	integer	ID della query (comando COPY) che ha generato questa riga.
line_number	integer	Quando viene specificata l'opzione COPY `SIMPLIFY AUTO`, questo valore è il numero di record del record semplificato nello shapefile.
maximum_tolerance	double	Il valore di tolleranza della distanza specificato nel comando COPY. Questo è il valore di tolleranza massima quando si utilizza l'opzione `SIMPLIFY AUTO` oppure il valore di tolleranza fissa quando si utilizza l'opzione `SIMPLIFY`.
initial_size	integer	La dimensione in byte del valore dei dati `GEOMETRY` prima della semplificazione.
simplified	char(1)	Quando viene specificata l'opzione COPY `SIMPLIFY AUTO`, `t` se la geometria è stata semplificata con successo, oppure `f` in caso contrario. La geometria potrebbe non essere semplificata correttamente se dopo la semplificazione con la tolleranza massima specificata, la sua dimensione è ancora maggiore della dimensione massima della geometria.
final_size	integer	Quando viene specificata l'opzione COPY `SIMPLIFY AUTO`, questa è la dimensione in byte della geometria dopo la semplificazione.

Nome colonna	Tipo di dati	Descrizione
final_tolerance	double	

Query di esempio

La seguente query restituisce l'elenco dei record semplificati da COPY.

```
SELECT * FROM svl_spatial_simplify WHERE query = pg_last_copy_id();
 query | line_number | maximum_tolerance | initial_size | simplified | final_size |
 final_tolerance
-------+-------------+-------------------+--------------+------------+------------
+----------------------
    20 |     1184704 |                -1 |      1513736 | t          |    1008808 |
 1.276386653895e-05
    20 |     1664115 |                -1 |      1233456 | t          |    1023584 |
 6.11707814796635e-06
```

SVL_SPECTRUM_SCAN_ERROR

È possibile eseguire query sulla vista di sistema SVL_SPECTRUM_SCAN_ERROR per ottenere informazioni sugli errori di scansione Redshift Spectrum.

SVL_SPECTRUM_SCAN_ERROR è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_EXTERNAL_QUERY_ERROR. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Visualizza un esempio di errori registrati. Il valore di default è 10 voci per query.

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente che ha generato questa riga.
query	integer	L'ID della query che ha generato questa riga.
posizione	character(128)	La posizione dei dati da interrogare.
rowid	character(128)	La posizione dell'errore all'interno del file. Le parti `rowid` sono separate da `:` (due punti); è possibile che altre parti vengano aggiunte in futuro. `row_offset :row_group :row_id` row_offset rappresenta l'offset (in byte) della riga all'interno del file ed è impostato su `-1` per i formati di file non supportati. Una tabella è divisa in row_groups e ogni gruppo include righe con row_id distinti.
colname	character(128)	Il nome della colonna restituito dalla query.
original_value	character(128)	Valore originale interrogato.
modified_value	character(128)	Valore modificato restituito in base all'opzione di configurazione di gestione dei dati specificata nella query.
Trigger	character(128)	Opzione di gestione dati specificata nella query.
action	character(128)	Azione associata all'opzione di gestione dei dati specificata nella query.
action_value	character(128)	Valore del parametro dell'operazione associato all'opzione di gestione dei dati specificata nella query.
error_code	integer	Codice risultato dell'opzione di gestione dati specificata nella query.

Query di esempio

La seguente query restituisce le righe per le quali sono state eseguite operazioni di gestione dei dati.

```
SELECT * FROM svl_spectrum_scan_error;
```

La query restituisce risultati simili a quanto esposto di seguito.

```
  userid  query     location                                    rowid   colname
         original_value             modified_value       trigger          action
                 action_valueerror_code
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:0     league_name
         Barclays Premier League    Barclays Premier Lea UNSPECIFIED      TRUNCATE
                                    156
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:0     league_nspi
          34595                     32767                UNSPECIFIED
OVERFLOW_VALUE                                  199
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:1     league_nspi
          34151                     32767                UNSPECIFIED
OVERFLOW_VALUE                                  199
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:2     league_name
         Barclays Premier League    Barclays Premier Lea UNSPECIFIED      TRUNCATE
                                    156
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:2     league_nspi
          33223                     32767                UNSPECIFIED
OVERFLOW_VALUE                                  199
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:3     league_name
         Barclays Premier League    Barclays Premier Lea UNSPECIFIED      TRUNCATE
                                    156
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:3     league_nspi
          32808                     32767                UNSPECIFIED
OVERFLOW_VALUE                                  199
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:4     league_nspi
          32790                     32767                UNSPECIFIED
OVERFLOW_VALUE                                  199
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:5     league_name
         Spanish Primera Division   Spanish Primera Divi UNSPECIFIED      TRUNCATE
                                    156
    100   1574007   s3://spectrum-uddh/league/spi_global_rankings.0:6     league_name
         Spanish Primera Division   Spanish Primera Divi UNSPECIFIED      TRUNCATE
                                    156
```

SVL_STATEMENTTEXT

Utilizzare la visualizzazione SVL_STATEMENTTEXT per ottenere un record completo di tutti i comandi SQL che sono stati eseguiti nel sistema.

La visualizzazione SVL_STATEMENTTEXT contiene l'unione di tutte le righe nelle tabelle STL_DDLTEXT, STL_QUERYTEXT e STL_UTILITYTEXT. Questa visualizzazione include anche un join alla tabella STL_QUERY.

SVL_STATEMENTTEXT è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	ID dell'utente che ha generato la voce.
xid	bigint	ID di transazione associato all'istruzione.
pid	integer	ID di processo per l'istruzione.
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etiche tta definita con un comando SET QUERY GROUP. Se la query non è basata su file o il parametro QUERY_GROUP non è impostato, questo campo è vuoto.
starttime	timestamp	Orario esatto in cui l'istruzione ha avviato l'esecuzione, con 6 cifre di precisione per le frazioni di secondo. Ad esempio: **2009-06-12 11:29:19.131358**

Nome colonna	Tipo di dati	Descrizione
endtime	timestamp	Orario esatto in cui l'istruzione ha terminato l'esecuzione, con 6 cifre di precisione per le frazioni di secondo. Ad esempio: **`2009-06-12 11:29:19.193640`**
sequenza	integer	Quando una singola istruzione contiene più di 200 caratteri , vengono registrate delle righe aggiuntive per tale istruzion e. La sequenza 0 è la prima riga, 1 la seconda e così via.
tipo	varchar(10)	Tipo di istruzione SQL: **`QUERY`**, **`DDL`** o **`UTILITY`**.
text	character(200)	Testo SQL, in incrementi da 200 caratteri. Questo campo potrebbe contenere caratteri speciali come barra rovesciat a (\\) e nuova riga (\n).

Query di esempio

La query seguente restituisce le istruzioni DDL eseguite il 16 giugno 2009:

```
select starttime, type, rtrim(text) from svl_statementtext
where starttime like '2009-06-16%' and type='DDL' order by starttime asc;

starttime                  | type |              rtrim
---------------------------|------|--------------------------------
2009-06-16 10:36:50.625097 | DDL  | create table ddltest(c1 int);
2009-06-16 15:02:16.006341 | DDL  | drop view alltickitjoin;
2009-06-16 15:02:23.65285  | DDL  | drop table sales;
2009-06-16 15:02:24.548928 | DDL  | drop table listing;
2009-06-16 15:02:25.536655 | DDL  | drop table event;
...
```

Ricostruzione dell'SDL archiviato

Per ricostruire l'SQL archiviato nella colonna `text` di SVL_STATEMENTTEXT, eseguire un'istruzione SELECT per creare SQL da 1 o più parti nella colonna `text`. Prima di eseguire l'SQL ricostruito, sostituire un carattere speciale qualsiasi (\n) con una nuova riga. Il risultato della dichiarazione SELECT seguente è dato da una riga di SQL ricostruiti nel campo `query_statement`.

```
select LISTAGG(CASE WHEN LEN(RTRIM(text)) = 0 THEN text ELSE RTRIM(text) END, '')
 within group (order by sequence) AS query_statement
from SVL_STATEMENTTEXT where pid=pg_backend_pid();
```

Ad esempio, la query seguente seleziona 3 colonne. La query stessa è più lunga di 200 caratteri ed è archiviata in parti in SVL_STATEMENTTEXT.

```
select
1 AS a012345678901234567890123456789012345678901234567890123456789012345678901234567890,
2 AS b012345678901234567890123456789012345678901234567890123456789012345678901234567890,
3 AS b012345678901234567890123456789012345678901234
FROM stl_querytext;
```

In questo esempio, la query è archiviata in 2 parti (righe) nella colonna `text` di SVL_STATEMENTTEXT.

```
select sequence, text from SVL_STATEMENTTEXT where pid = pg_backend_pid() order by
 starttime, sequence;
```

```
 sequence |
                 text

----------
+-----------------------------------------------------------------------------------------------------
        0 | select\n1 AS
 a012345678901234567890123456789012345678901234567890123456789012345678901234567890,\n2 AS
 b012345678901234567890123456789012345678901234567890123456789012345678901234567890,\n3 AS
 b012345678901234567890123456789012345678901234
        1 | \nFROM stl_querytext;
```

Per ricostruire l'SQL archiviato in STL_STATEMENTTEXT, eseguire l'SQL seguente.

```
select LISTAGG(CASE WHEN LEN(RTRIM(text)) = 0 THEN text ELSE RTRIM(text) END, '')
 within group (order by sequence) AS text
from SVL_STATEMENTTEXT where pid=pg_backend_pid();
```

Per utilizzare l'SQL ricostruito derivante nel client, sostituire qualsiasi carattere speciale (\n) con una nuova riga.

```
                    text

----------------------------------------------------------------------------------------------------
 select\n1 AS a012345678901234567890123456789012345678901234567890123456789012345678901234567890,
\n2 AS b012345678901234567890123456789012345678901234567890123456789012345678901234567890,\n3 AS
 b012345678901234567890123456789012345678901234\nFROM stl_querytext;
```

SVL_STORED_PROC_CALL

Puoi eseguire delle query sulla visualizzazione del sistema SVL_STORED_PROC_CALL per ottenere informazioni sulle chiamate di procedura archiviata, tra cui l'ora di inizio, l'ora di fine e se una chiamata viene annullata. Ogni chiamata di procedura archiviata riceve un'ID query.

SVL_STORED_PROC_CALL è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_PROCEDURE_CALL. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente i quali privilegi erano stati utilizzati per eseguire l'istruzione. Se questa chiamata è stata nidificata in una procedura archiviata SECURITY DEFINER, questo è l'ID utente del proprietario di quella procedura archiviata.
session_userid	integer	L'ID dell'utente che ha creato la sessione ed è l'interme diario della chiamata di procedura archiviata di primo livello.
query	integer	L'ID query della chiamata di procedura.

Nome colonna	Tipo di dati	Descrizione
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etiche tta definita con un comando SET QUERY GROUP. Se la query non è basata su file o non è impostato il parametro QUERY_GROUP, questo valore del campo è predefinito.
xid	bigint	L'ID transazione.
pid	integer	L'ID di processo. In genere, tutte le query in una sessione sono eseguite nello stesso processo, quindi questo valore di solito rimane costante se esegui una serie di query nella stessa sessione. Seguendo determinati eventi interni, Amazon Redshift può riavviare una sessione attiva e assegnare un nuovo valore PID. Per ulteriori informazioni, consulta STL_RESTARTED_SESSIONS.
database	character(32)	The name of the database that the user was connected to when the query was issued.
querytxt	character(4000)	Il testo effettivo della query della chiamata di procedura.
starttime	timestamp	L'orario in UTC in cui la query ha avviato l'esecuzione, con sei cifre di precisione per le frazioni di secondo, ad esempio: **2009-06-12 11:29:19.131358.**
endtime	timestamp	L'orario in UTC in cui la query ha terminato l'esecuzione, con sei cifre di precisione per le frazioni di secondo, ad esempio: **2009-06-12 11:29:19.131358.**
aborted	integer	Se una procedura archiviata è stata interrotta dal sistema o annullata dall'utente, questa colonna contiene 1. Se la chiamata è stata completata, questa colonna contiene 0.
from_sp_call	integer	Se la chiamata di procedura è stata richiamata da un'altra chiamata di procedura, questa colonna contiene l'ID query della chiamata esterna. Altrimenti il campo è NULL.

Query di esempio

La seguente query restituisce il tempo trascorso in ordine decrescente e lo stato di completamento per le chiamate di procedure archiviate nel giorno precedente.

```
select query, datediff(seconds, starttime, endtime) as elapsed_time, aborted,
 trim(querytxt) as call from svl_stored_proc_call where starttime >= getdate() -
 interval '1 day' order by 2 desc;

  query | elapsed_time | aborted |                                          call
--------+--------------+---------
+-------------------------------------------------------------------------------
   4166 |            7 |       0 | call search_batch_status(35,'succeeded');
   2433 |            3 |       0 | call test_batch (123456)
   1810 |            1 |       0 | call prod_benchmark (123456)
   1836 |            1 |       0 | call prod_testing (123456)
   1808 |            1 |       0 | call prod_portfolio ('N', 123456)
   1816 |            1 |       1 | call prod_portfolio ('Y', 123456)
```

SVL_STORED_PROC_MESSAGES

Puoi eseguire query sulla vista di sistema SVL_STORED_PROC_MESSAGES per ottenere informazioni sui messaggi delle procedure archiviate. I messaggi generati vengono registrati anche se la chiamata alla procedura archiviata viene annullata. Ogni chiamata di procedura archiviata riceve un'ID query. Per ulteriori informazioni su come impostare il livello minimo per i messaggi registrati, consultare stored_proc_log_min_messages.

SVL_STORED_PROC_MESSAGES è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_PROCEDURE_MESSAGES. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
userid	integer	L'ID dell'utente i quali privilegi erano stati utilizzati per eseguire l'istruzione. Se questa chiamata è stata nidificata in una procedura archiviata SECURITY DEFINER, questo è l'ID utente del proprietario di quella procedura archiviata.
session_userid	integer	L'ID dell'utente che ha creato la sessione ed è l'interme diario della chiamata di procedura archiviata di primo livello.
pid	integer	L'ID di processo.
xid	bigint	L'ID della transazione della query della chiamata di procedura.
query	integer	L'ID query della chiamata di procedura.
recordtime	timestamp	L'ora di generazione del messaggio in UTC.
loglevel	integer	Il valore numerico del livello di log del messaggio generato. Valori possibili: 20 – per LOG 30 – per INFO 40 – per NOTICE 50 – per WARNING 60 – per EXCEPTION
loglevel_text	character(10)	Il livello di log che corrisponde al valore numerico in loglevel. Valori possibili: LOG, INFO, NOTICE, WARNING ed EXCEPTION.
message	character(1024)	Il testo del messaggio generato.
linenum	integer	Il numero di riga dell'istruzione generata.
querytext	character(500)	Il testo effettivo della query della chiamata di procedura.
etichetta	character(320)	Il nome del file utilizzato per eseguire la query o un'etiche tta definita con un comando SET QUERY GROUP. Se la

Nome colonna	Tipo di dati	Descrizione
		query non è basata su file o non è impostato il parametro QUERY_GROUP, questo valore del campo è predefinito.
aborted	integer	Se una procedura archiviata è stata interrotta dal sistema o annullata dall'utente, questa colonna contiene 1. Se la chiamata è stata completata, questa colonna contiene 0.
message_xid	bigint	L'ID della transazione del messaggio generato.

Query di esempio

Le seguenti istruzioni SQL mostrano come utilizzare SVL_STORED_PROC_MESSAGES per esaminare i messaggi generati.

```
-- Create and run a stored procedure
CREATE OR REPLACE PROCEDURE test_proc1(f1 int) AS
$$
BEGIN
    RAISE INFO 'Log Level: Input f1 is %',f1;
    RAISE NOTICE 'Notice Level: Input f1 is %',f1;
    EXECUTE 'select invalid';
    RAISE NOTICE 'Should not print this';

EXCEPTION WHEN OTHERS THEN
     raise exception 'EXCEPTION level: Exception Handling';
END;
$$ LANGUAGE plpgsql;

-- Call this stored procedure
CALL test_proc1(2);

-- Show raised messages with level higher than INFO
SELECT query, recordtime, loglevel, loglevel_text, trim(message) as message, aborted
 FROM svl_stored_proc_messages
  WHERE loglevel > 30 AND query = 193 ORDER BY recordtime;
```

```
 query |         recordtime         | loglevel | loglevel_text |               message
              | aborted
-------+----------------------------+----------+---------------
+-----------------------------------+---------
   193 | 2020-03-17 23:57:18.277196 |       40 | NOTICE        | Notice Level: Input f1
 is 2          |       1
   193 | 2020-03-17 23:57:18.277987 |       60 | EXCEPTION     | EXCEPTION level:
 Exception Handling |       1
(2 rows)

-- Show raised messages at EXCEPTION level
SELECT query, recordtime, loglevel, loglevel_text, trim(message) as message, aborted
 FROM svl_stored_proc_messages
  WHERE loglevel_text = 'EXCEPTION' AND query = 193 ORDER BY recordtime;

 query |         recordtime         | loglevel | loglevel_text |               message
              | aborted
-------+----------------------------+----------+---------------
+-----------------------------------+---------
   193 | 2020-03-17 23:57:18.277987 |       60 | EXCEPTION     | EXCEPTION level:
 Exception Handling |       1
```

Le seguenti istruzioni SQL mostrano come utilizzare SVL_STORED_PROC_MESSAGES per esaminare i messaggi generati con l'opzione SET durante la creazione di una procedura archiviata. Poiché test_proc() dispone di un livello di log minimo di NOTICE, solo i messaggi di livello NOTICE, WARNING ed EXCEPTION vengono registrati in SVL_STORED_PROC_MESSAGES.

```
-- Create a stored procedure with minimum log level of NOTICE
CREATE OR REPLACE PROCEDURE test_proc() AS
$$
BEGIN
    RAISE LOG 'Raise LOG messages';
    RAISE INFO 'Raise INFO messages';
    RAISE NOTICE 'Raise NOTICE messages';
    RAISE WARNING 'Raise WARNING messages';
    RAISE EXCEPTION 'Raise EXCEPTION messages';
    RAISE WARNING 'Raise WARNING messages again'; -- not reachable
END;
$$ LANGUAGE plpgsql SET stored_proc_log_min_messages = NOTICE;

-- Call this stored procedure
CALL test_proc();
```

```
-- Show the raised messages
SELECT query, recordtime, loglevel_text, trim(message) as message, aborted FROM
 svl_stored_proc_messages
  WHERE query = 149 ORDER BY recordtime;

 query |         recordtime         | loglevel_text |          message         |
 aborted
-------+----------------------------+---------------+--------------------------
+---------
   149 | 2020-03-16 21:51:54.847627 | NOTICE        | Raise NOTICE messages    |
 1
   149 | 2020-03-16 21:51:54.84766  | WARNING       | Raise WARNING messages   |
 1
   149 | 2020-03-16 21:51:54.847668 | EXCEPTION     | Raise EXCEPTION messages |
 1
(3 rows)
```

SVL_TERMINATE

Registra l'ora in cui un utente annulla o termina un processo.

SELECT PG_TERMINATE_BACKEND(pid), SELECT PG_CANCEL_BACKEND(pid) e CANCEL pid crea una voce di log in SVL_TERMINATE.

SVL_TERMINATE è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_QUERY_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
pid	integer	L'ID processo del processo annullato o terminato.
eventtime	timestamp	L'ora in cui il processo viene annullato o terminato.
userid	integer	L'ID utente dell'utente che esegue il comando.

Nome colonna	Tipo di dati	Descrizione
tipo	string	Il tipo di terminazione. Può essere CANCEL o TERMINATE.

Il seguente comando mostra l'ultima query annullata.

```
select * from svl_terminate order by eventtime desc limit 1;
 pid  |         eventtime          | userid |  type
------+----------------------------+--------+--------
 8324 | 2020-03-24 09:42:07.298937 |      1 | CANCEL
(1 row)
```

SVL_UDF_LOG

Registra la generazione di messaggi di errore e di avviso definita dal sistema durante l'esecuzione della funzione definita dall'utente (UDF).

SVL_UDF_LOG è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_UDF_LOG. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
query	bigint	L'ID di query. È possibile utilizzare questo ID per unire varie altre tabelle e visualizz azioni di sistema.
message	char(4096)	Il messaggio generato dalla funzione.

Nome colonna	Tipo di dati	Descrizione
creato	timestamp	L'orario in cui è stato creato il log.
traceback	char(4096)	Se disponibile, questo valore fornisce un'analisi dello stack per l'UDF. Per ulteriori informazioni, consultare analisi in Python Standard Library.
funcname	character(256)	Il nome dell'UDF in esecuzione.
nodo	integer	Il nodo in cui il messaggio è stato generato.
sezione	integer	La sezione in cui il messaggio è stato generato.
seq	integer	La sequenza del messaggio sulla sezione.

Query di esempio

L'esempio seguente mostra come le UDF gestiscono gli errori definiti dal sistema. Il primo blocco mostra la definizione di una funzione UDF che restituisce l'inverso di un argomento. Quando si esegue la funzione e si fornisce un argomento 0, come mostra il secondo blocco, la funzione restituisce un errore. La terza istruzione legge il messaggio di errore registrato in SVL_UDF_LOG

```
-- Create a function to find the inverse of a number

CREATE OR REPLACE FUNCTION  f_udf_inv(a int)
  RETURNS float IMMUTABLE
AS $$
   return 1/a
$$ LANGUAGE plpythonu;

-- Run the function with a 0 argument to create an error
Select f_udf_inv(0) from sales;
```

```
-- Query SVL_UDF_LOG to view the message

Select query, created, message::varchar
from svl_udf_log;

 query |          created           | message
-------+----------------------------
+-----------------------------------------------------------
  2211 | 2015-08-22 00:11:12.04819  | ZeroDivisionError: long division or modulo by
 zero\nNone
```

L'esempio seguente aggiunge la registrazione e un messaggio di avviso all'UDF in modo che un'operazione di divisione per zero risulti in un messaggio di avviso anziché in un arresto con un messaggio di errore.

```
-- Create a function to find the inverse of a number and log a warning

CREATE OR REPLACE FUNCTION f_udf_inv_log(a int)
  RETURNS float IMMUTABLE
 AS $$
  import logging
  logger = logging.getLogger() #get root logger
  if a==0:
    logger.warning('You attempted to divide by zero.\nReturning zero instead of error.
\n')
    return 0
  else:
     return 1/a
$$ LANGUAGE plpythonu;
```

L'esempio seguente esegue la funzione, poi fa una query a SVL_UDF_LOG per visualizzare il messaggio.

```
-- Run the function with a 0 argument to trigger the warning
Select f_udf_inv_log(0) from sales;

-- Query SVL_UDF_LOG to view the message

Select query, created, message::varchar
from svl_udf_log;
```

```
query |          created           | message
------+----------------------------+---------------------------------
    0 | 2015-08-22 00:11:12.04819  | You attempted to divide by zero.
                                     Returning zero instead of error.
```

SVL_USER_INFO

Puoi recuperare i dati sugli utenti del database Amazon Redshift con la vista SVL_USER_INFO.

SVL_USER_INFO è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
nome utente	text	Il nome utente del ruolo.
usesysid	integer	l'ID utente dell'utente.
usecreatedb	booleano	Un valore che indica se l'utente dispone delle autorizzazioni per creare i database.
usesuper	booleano	Un valore che indica se l'utente è un utente con privilegi avanzati.
usecatupd	booleano	Un valore che indica se l'utente può aggiornare i cataloghi di sistema.
useconnlimit	text	Il numero di connessioni che l'utente può aprire.
syslogaccess	text	Un valore che indica se l'utente dispone dell'acceso ai log di sistema. I due possibili valori sono `RESTRICTED` e `UNRESTRICTED`. `RESTRICTED` indica che gli utenti che non sono utenti con privilegi avanzati possono visualizzare i propri record. `UNRESTRICTED` indica che gli utenti che non sono utenti con privilegi avanzati possono visualizzare tutti i record nelle viste e nelle tabelle di sistema di cui possiedono i privilegi `SELECT`.

Nome colonna	Tipo di dati	Descrizione
last_ddl_ts	timestamp	Il timestamp per l'ultima istruzione di creazione data definition language (DDL) eseguita dall'utente.
sessionti meout	integer	Il tempo massimo in secondi in cui una sessione rimane inattiva o inattiva prima del timeout. 0 indica che non è impostato alcun timeout. Per informazioni sull'impostazione del timeout inattivo del cluster, consulta Quote e limiti in Amazon Redshift nella Guida alla gestione di Amazon Redshift.
external_ id	text	L'identificatore univoco dell'utente nel provider di identità di terze parti.

Query di esempio

Il comando seguente recupera le informazioni sull'utente da SVL_USER_INFO.

```
SELECT * FROM SVL_USER_INFO;
```

SVL_VACUUM_PERCENTAGE

La visualizzazione SVL_VACUUM_PERCENTAGE riporta la percentuale di blocchi di dati allocati a una tabella dopo aver eseguito un vacuum. Questo numero percentuale mostra quanto spazio su disco è stato recuperato. Consultare il comando VACUUM per ulteriori informazioni sull'utilità vacuum.

SVL_VACUUM_PERCENTAGE è visibile solo per gli utenti con privilegi avanzati. Per ulteriori informazioni, consulta Visibilità dei dati nelle tabelle e nelle viste di sistema.

Alcuni o tutti i dati di questa tabella sono definiti anche nella vista di monitoraggio SYS SYS_VACUUM_HISTORY. I dati nella vista di monitoraggio SYS sono formattati in modo da essere più facili da usare e comprendere. Ti consigliamo di utilizzare la vista di monitoraggio SYS per le query.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
xid	bigint	ID di transazione per l'istruzione di vacuum.
table_id	integer	ID di tabella per la tabella sottoposta a vacuum.
percentage	bigint	Percentuale di blocchi di dati dopo un vacuum (relativa al numero di blocchi nella tabella prima dell'esecuzione del vacuum).

Query di esempio

La query seguente visualizza la percentuale di una determinata operazione sulla tabella 100238:

```
select * from svl_vacuum_percentage
where table_id=100238 and xid=2200;

xid  | table_id | percentage
-----+----------+------------
1337 |   100238 |         60
(1 row)
```

Dopo questa operazione di vacuum, la tabella conteneva il 60 percento dei blocchi originali.

Tabelle di catalogo di sistema

Argomenti

- PG_ATTRIBUTE_INFO
- PG_CLASS_INFO
- PG_DATABASE_INFO
- PG_DEFAULT_ACL
- PG_EXTERNAL_SCHEMA
- PG_LIBRARY
- PG_PROC_INFO
- PG_STATISTIC_INDICATOR

- PG_TABLE_DEF
- PG_USER_INFO
- Query sulle tabelle di catalogo

Nei cataloghi di sistema vengono archiviati i metadati degli schemi, ad esempio le informazioni su tabelle e colonne. Le tabelle di catalogo di sistema hanno il prefisso PG.

Gli utenti di Amazon Redshift possono accedere alle tabelle di catalogo PostgreSQL standard. Per ulteriori informazioni sui cataloghi di sistema di PostgreSQL, consultare la pagina relativa alle tabelle di sistema di PostgreSQL.

PG_ATTRIBUTE_INFO

PG_ATTRIBUTE_INFO è una vista di sistema di Amazon Redshift costruita sulla tabella di catalogo PostgreSQL PG_ATTRIBUTE e sulla tabella di catalogo interna PG_ATTRIBUTE_ACL. PG_ATTRIBUTE_INFO include i dettagli sulle colonne di una tabella o vista, incluse le liste di controllo accessi delle colonne, se presenti.

Colonne di tabella

PG_ATTRIBUTE_INFO mostra la colonna seguente oltre alle colonne in PG_ATTRIBUTE.

Nome colonna	Tipo di dati	Descrizione
attacl	aclitem[]	I privilegi di accesso a livello di colonna, se presenti, che sono stati concessi specificamente in questa colonna.

PG_CLASS_INFO

PG_CLASS_INFO è una vista di sistema di Amazon Redshift integrata sulle tabelle di catalogo PostgreSQL PG_CLASS e PG_CLASS_EXTENDED. PG_CLASS_INFO include dettagli sull'ora di creazione della tabella e sullo stile di distribuzione attuale. Per ulteriori informazioni, consultare Utilizzo degli stili di distribuzione dati.

PG_CLASS_INFO è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

PG_CLASS_INFO mostra le seguenti colonne in aggiunta alle colonne in PG_CLASS. La colonna `oid` in PG_CLASS è denominata `relid` nella tabella PG_CLASS_INFO.

Nome colonna	Tipo di dati	Descrizione
relcreationtime	timestamp	Ora in UTC in cui la tabella è stata creata.
releffectivediststyle	integer	Lo stile di distribuzione di una tabella o, se la tabella utilizza la distribuzione automatica, lo stile di distribuzione attuale assegnato da Amazon Redshift.

La colonna RELEFFECTIVEDISTSTYLE in PG_CLASS_INFO indica lo stile di distribuzione attuale per la tabella. Se la tabella utilizza la distribuzione automatica, RELEFFECTIVEDISTSTYLE è 10, 11 o 12 che indica se lo stile di distribuzione effettivo è AUTO (ALL), AUTO (EVEN) o AUTO (KEY). Se la tabella utilizza la distribuzione automatica, lo stile di distribuzione potrebbe inizialmente mostrare AUTO (ALL), quindi passare ad AUTO (EVEN) quando del dimensioni della tabella aumentano o AUTO (KEY) se una colonna viene considerata utile come chiave di distribuzione.

La seguente tabella fornisce lo stile di distribuzione per ogni valore su RELEFFECTIVEDISTSTYLE:

RELEFFECTIVEDISTSTYLE	Stile di distribuzione attuale
0	EVEN
1	KEY
8	ALL
10	AUTO (ALL)
11	AUTO (EVEN)
12	AUTO (KEY)

Esempio

La query seguente restituisce lo stile di distribuzione corrente delle tabelle nel catalogo.

```
select reloid as tableid,trim(nspname) as schemaname,trim(relname) as
 tablename,reldiststyle,releffectivediststyle,
CASE WHEN "reldiststyle" = 0 THEN 'EVEN'::text
     WHEN "reldiststyle" = 1 THEN 'KEY'::text
     WHEN "reldiststyle" = 8 THEN 'ALL'::text
     WHEN "releffectivediststyle" = 10 THEN 'AUTO(ALL)'::text
     WHEN "releffectivediststyle" = 11 THEN 'AUTO(EVEN)'::text
     WHEN "releffectivediststyle" = 12 THEN 'AUTO(KEY)'::text ELSE '<<UNKNOWN>>'::text
 END as diststyle,relcreationtime
from pg_class_info a left join pg_namespace b on a.relnamespace=b.oid;
```

```
 tableid | schemaname | tablename | reldiststyle | releffectivediststyle | diststyle  |
      relcreationtime
---------+------------+-----------+--------------+-----------------------+------------
+----------------------------
 3638033 | public     | customer  |            0 |                     0 | EVEN       |
 2019-06-13 15:02:50.666718
 3638037 | public     | sales     |            1 |                     1 | KEY        |
 2019-06-13 15:03:29.595007
 3638035 | public     | lineitem  |            8 |                     8 | ALL        |
 2019-06-13 15:03:01.378538
 3638039 | public     | product   |            9 |                    10 | AUTO(ALL)  |
 2019-06-13 15:03:42.691611
 3638041 | public     | shipping  |            9 |                    11 | AUTO(EVEN) |
 2019-06-13 15:03:53.69192
 3638043 | public     | support   |            9 |                    12 | AUTO(KEY)  |
 2019-06-13 15:03:59.120695
(6 rows)
```

PG_DATABASE_INFO

PG_DATABASE_INFO è una vista di sistema Amazon Redshift che estende la tabella di catalogo PostgreSQL PG_DATABASE.

PG_DATABASE_INFO è visibile a tutti gli utenti.

Colonne di tabella

PG_DATABASE_INFO contiene le seguenti colonne oltre alle colonne in PG_DATABASE. La colonna `oid` in PG_DATABASE è denominata `datid` nella tabella PG_DATABASE_INFO. Per ulteriori informazioni, consultare la documentazione di PostgreSQL.

Nome colonna	Tipo di dati	Descrizione
datid	oid	L'identificatore di oggetto (OID) utilizzato internamente dalle tabelle di sistema.
datconnlimit	text	Il numero massimo di connessioni simultanee che possono essere effettuate a questo database. Il valore -1 indica nessun limite.

PG_DEFAULT_ACL

Archivia le informazioni sui privilegi di accesso predefiniti. Per ulteriori informazioni sui privilegi di accesso predefiniti, consultare ALTER DEFAULT PRIVILEGES.

PG_DEFAULT_ACL è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
defacluser	integer	ID dell'utente a cui si applicano i privilegi elencati.
defaclnamespace	oid	ID oggetto dello schema a cui si applicano i privilegi predefiniti. Se non è specificato alcuno schema, il valore predefinito è 0.
defaclobjtype	carattere	Tipo di oggetto al quale si applicano i privilegi predefiniti. I valori validi sono:

Nome colonna	Tipo di dati	Descrizione
		• r–relazione (tabella o vista) • f–function • p–stored procedure

Nome colonna	Tipo di dati	Descrizione
defaclacl	aclitem[]	Stringa che definisce i privilegi predefiniti per l'utente o il gruppo di utenti specificato e il tipo di oggetto. Se i privilegi vengono concessi a un utente, la stringa ha il formato seguente: *{ username=privilegestring/grantor }* username Nome dell'utente a cui vengono concessi i privilegi. Se si omette username, i privilegi vengono concessi a PUBLIC. Se i privilegi vengono concessi a un gruppo di utenti, la stringa ha il formato seguente: *{ "group groupname=privilegestring/grantor" }* privilegestring Stringa che specifica i privilegi concessi. I valori validi sono: • r–SELECT (lettura) • a–INSERT (aggiunta) • w–UPDATE (scrittura) • d–DELETE • x-Concede il privilegio di creazione di un vincolo di chiave esterna (REFERENCES). • X—ESEGUI • *Indica che l'utente che riceve il privilegio precedente può a sua volta concedere lo stesso privilegio ad altri (WITH GRANT OPTION).

Nome colonna	Tipo di dati	Descrizione
		grantor Nome dell'utente che ha concesso i privilegi. L'esempio seguente indica che l'utente `admin` ha concesso tutti i privilegi, incluso WITH GRANT OPTION, all'utente `dbuser`. `dbuser=r*a*w*d*x*X*/admin`

Esempio

La query seguente restituisce tutti i privilegi predefiniti definiti per il database.

```
select pg_get_userbyid(d.defacluser) as user,
n.nspname as schema,
case d.defaclobjtype when 'r' then 'tables' when 'f' then 'functions' end
as object_type,
array_to_string(d.defaclacl, ' + ')  as default_privileges
from pg_catalog.pg_default_acl d
left join pg_catalog.pg_namespace n on n.oid = d.defaclnamespace;

 user  | schema | object_type |              default_privileges
-------+--------+-------------+-----------------------------------------------------
 admin | tickit | tables      | user1=r/admin + "group group1=a/admin" + user2=w/admin
```

Il risultato dell'esempio precedente mostra che per tutte le nuove tabelle create dall'utente `admin` nello schema `tickit`, `admin` concede i privilegi SELECT a `user1`, i privilegi INSERT a `group1` e i privilegi UPDATE a `user2`.

PG_EXTERNAL_SCHEMA

Archivia le informazioni sugli schemi esterni.

PG_EXTERNAL_SCHEMA è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti normali visualizzano solo i metadati a cui hanno accesso. Per ulteriori informazioni, consultare CREATE EXTERNAL SCHEMA.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
esoid	oid	ID dello schema esterno.
eskind	integer	Tipo di schema esterno.
esdbname	text	Nome del database esterno.
esoptions	text	Opzioni dello schema esterno.

Esempio

Nell'esempio seguente vengono visualizzati i dettagli per gli schemi esterni.

```
select esoid, nspname as schemaname, nspowner, esdbname as external_db, esoptions
from pg_namespace a,pg_external_schema b where a.oid=b.esoid;

esoid  | schemaname      | nspowner | external_db | esoptions

-------+-----------------+----------+-------------
+-------------------------------------------------------------
100134 | spectrum_schema |      100 | spectrum_db |
 {"IAM_ROLE":"arn:aws:iam::123456789012:role/mySpectrumRole"}
100135 | spectrum        |      100 | spectrumdb  |
 {"IAM_ROLE":"arn:aws:iam::123456789012:role/mySpectrumRole"}
100149 | external        |      100 | external_db |
 {"IAM_ROLE":"arn:aws:iam::123456789012:role/mySpectrumRole"}
```

PG_LIBRARY

Archivia le informazioni sulle librerie definite dall'utente.

PG_LIBRARY è visibile a tutti gli utenti. Gli utenti con privilegi avanzati visualizzano tutte le righe; gli utenti regolari visualizzano solo i propri dati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
name	name	Nome della libreria.
language_oid	oid	Riservate per il sistema.
file_store_id	integer	Riservate per il sistema.
owner	integer	ID utente del proprietario della libreria.

Esempio

L'esempio seguente restituisce informazioni sulle librerie installate dall'utente.

```
select * from pg_library;

name       | language_oid | file_store_id | owner
-----------+--------------+---------------+------
f_urlparse |       108254 |          2000 |   100
```

PG_PROC_INFO

PG_PROC_INFO è una vista di sistema di Amazon Redshift costruita sulla tabella di catalogo PostgreSQL PG_PROC e sulla tabella di catalogo interna PG_PROC_EXTENDED. PG_PROC_INFO include dettagli sulle procedure e funzioni archiviate, comprese le informazioni relative agli argomenti di output, se presenti.

Colonne di tabella

PG_PROC_INFO mostra le seguenti colonne in aggiunta alle colonne in PG_PROC. La colonna `oid` in PG_PROC è denominata `prooid` nella tabella PG_PROC_INFO.

Nome colonna	Tipo di dati	Descrizione
prooid	oid	L'ID oggetto della funzione o procedura archiviata.
prokind	"char"	Un valore che indica il tipo di funzioni o procedure archiviate. Questo valore è ‘f' per le funzioni regolari, ‘p' per le procedure archiviate e ‘a' per funzioni aggregate.
proargmodes	"char"[ ]	Un array con le modalità degli argomenti di procedura, codificati come ‘i' per argomenti IN, ‘o' per argomenti OUT e ‘b' per argomenti INOUT. Se tutti gli argomenti sono argomenti IN, il campo è NULL. I pedici corrispondono alle posizione nell'array proallargtypes.
proallargtypes	oid[ ]	Un array con i tipi di dati degli argomenti della procedura. Questo array include tutti i tipi di argomenti (compresi gli argomenti OUT e INOUT). Tuttavia, se tutti gli argomenti sono argomenti IN, questo campo è NULL. Il pedice è basato su uno. Invece il pedice su proargtypes viene effettuato da zero.

Il campo proargnames in PG_PROC_INFO contiene i nomi di tutti i tipi di argomenti (compresi OUT e INOUT), se presenti.

PG_STATISTIC_INDICATOR

Archivia le informazioni sul numero di righe inserite o eliminate dall'ultima operazione ANALYZE. La tabella PG_STATISTIC_INDICATOR viene aggiornata di frequente in seguito alle operazioni DML, quindi le statistiche sono approssimative.

PG_STATISTIC_INDICATOR è visibile solo agli utenti con privilegi avanzati. Per ulteriori informazioni, consultare Visibilità dei dati nelle tabelle e nelle viste di sistema.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
stairelid	oid	ID tabella
stairows	float	Numero totale di righe nella tabella.
staiins	float	Numero di righe inserite dall'ultima operazione ANALYZE.
staidels	float	Numero di righe eliminate o aggiornate dall'ultima operazione ANALYZE.

Esempio

L'esempio seguente restituisce informazioni per le modifiche alla tabella dall'ultima operazione ANALYZE.

```
select * from pg_statistic_indicator;

stairelid | stairows | staiins | staidels
----------+----------+---------+---------
   108271 |       11 |       0 |        0
   108275 |      365 |       0 |        0
   108278 |     8798 |       0 |        0
   108280 |    91865 |       0 |   100632
   108267 |    89981 |   49990 |     9999
   108269 |      808 |     606 |      374
   108282 |   152220 |   76110 |   248566
```

PG_TABLE_DEF

Archivia le informazioni sulle colonne di tabella.

PG_TABLE_DEF restituisce solo le informazioni sulle tabelle visibili all'utente. Se PG_TABLE_DEF non restituisce i risultati previsti, verifica che il parametro search_path sia impostato correttamente per includere gli schemi rilevanti.

Puoi usare SVV_TABLE_INFO per visualizzare informazioni più complete su una tabella, incluse differenze di distribuzione dei dati, differenze di distribuzione delle chiavi, dimensioni della tabella e statistiche.

Colonne di tabella

Nome colonna	Tipo di dati	Descrizione
schemaname	name	Nome schema.
tablename	name	Nome tabella.
column	name	Nome della colonna.
type	text	Tipo di dati della colonna.
encoding	character(32)	Codifica della colonna.
distkey	booleano	VERO se la colonna corrisponde alla chiave di distribuzione per la tabella.
sortkey	integer	Ordine della colonna nella chiave di ordinamento. Se la tabella usa una chiave di ordinamento composta, tutte le colonne che fanno parte della chiave di ordinamento hanno un valore positivo che indica la posizione della colonna nella chiave di ordinamento. Se la tabella usa una chiave di ordinamento con interlacciamento, ogni colonna che fa parte della chiave di ordinamento ha un valore alternativamente positivo o negativo, il cui valore assoluto indica la posizione della colonna nella chiave di ordinamento. Se il valore è 0, la colonna non fa parte di una chiave di ordinamento.
notnull	booleano	True se la colonna ha un vincolo NOT NULL.

Esempio

L'esempio seguente mostra le colonne della chiave di ordinamento composta per la tabella LINEORDER_COMPOUND.

```
select "column", type, encoding, distkey, sortkey, "notnull"
from pg_table_def
where tablename = 'lineorder_compound'
and sortkey <> 0;

column       | type    | encoding | distkey | sortkey | notnull
-------------+---------+----------+---------+---------+--------
lo_orderkey  | integer | delta32k | false   |       1 | true
lo_custkey   | integer | none     | false   |       2 | true
lo_partkey   | integer | none     | true    |       3 | true
lo_suppkey   | integer | delta32k | false   |       4 | true
lo_orderdate | integer | delta    | false   |       5 | true
(5 rows)
```

L'esempio seguente mostra le colonne della chiave di ordinamento con interfoliazione per la tabella LINEORDER_INTERLEAVED.

```
select "column", type, encoding, distkey, sortkey, "notnull"
from pg_table_def
where tablename = 'lineorder_interleaved'
and sortkey <> 0;

column       | type    | encoding | distkey | sortkey | notnull
-------------+---------+----------+---------+---------+--------
lo_orderkey  | integer | delta32k | false   |      -1 | true
lo_custkey   | integer | none     | false   |       2 | true
lo_partkey   | integer | none     | true    |      -3 | true
lo_suppkey   | integer | delta32k | false   |       4 | true
lo_orderdate | integer | delta    | false   |      -5 | true
(5 rows)
```

PG_TABLE_DEF restituisce solo le informazioni per le tabelle negli schemi che fanno parte del percorso di ricerca. Per ulteriori informazioni, consultare search_path.

Supponi, ad esempio, di creare un nuovo schema e una nuova tabella e quindi di eseguire una query su PG_TABLE_DEF.

```
create schema demo;
create table demo.demotable (one int);
select * from pg_table_def where tablename = 'demotable';

schemaname|tablename|column| type | encoding | distkey | sortkey | notnull
----------+---------+------+------+----------+---------+---------+--------
```

La query non restituisce righe per la nuova tabella. Esamina l'impostazione di `search_path`.

```
show search_path;

  search_path
---------------
 $user, public
(1 row)
```

Aggiungi lo schema demo al percorso di ricerca ed esegui di nuovo la query.

```
set search_path to '$user', 'public', 'demo';

select * from pg_table_def where tablename = 'demotable';

schemaname| tablename |column|  type   | encoding |distkey|sortkey| notnull
----------+-----------+------+---------+----------+-------+-------+--------
demo      | demotable | one  | integer | none     | f     |     0 | f
(1 row)
```

PG_USER_INFO

PG_USER_INFO è una vista di sistema Amazon Redshift che mostra le informazioni sull'utente, come l'ID utente e la scadenza della password.

Solo i superutenti possono vedere PG_USER_INFO.

Colonne di tabella

PG_USER_INFO contiene le seguenti colonne. Per ulteriori informazioni, consultare la documentazione di PostgreSQL.

Nome colonna	Tipo di dati	Descrizione
nome utente	name	Il nome utente.
usesysid	integer	L'ID utente.
usecreatedb	booleano	True se l'utente può creare database.
usesuper	booleano	True se l'utente è un superutente.
usecatupd	booleano	Se true, indica che l'utente può aggiornare i cataloghi di sistema.
passwd	text	La password.
valuntil	abstime	Data e ora di scadenza della password.
useconfig	text[]	La sessione usa in modo predefinito le variabili di runtime.
useconnlimit	text	Il numero di connessioni che l'utente può aprire.

Query sulle tabelle di catalogo

Argomenti

- Esempi di query di catalogo

In generale, è possibile unire viste e tabelle di catalogo (relazioni i cui nomi iniziano con **PG_**) a viste e tabelle di Amazon Redshift.

Le tabelle di catalogo usano un numero di tipi di dati non supportati da Amazon Redshift. I tipi di dati seguenti sono supportati quando le query uniscono tabelle di catalogo alle tabelle di Amazon Redshift:

- bool

- "char"
- float4
- int2
- int4
- int8
- name
- oid
- text
- varchar

Se pertanto scrivi una query di unione che fa riferimento in modo esplicito o implicito a una colonna contenente un tipo di dati non supportato, la query restituisce un errore. Anche le funzioni SQL usate in alcune tabelle di catalogo non sono supportate, ad eccezione di quelle usate dalle tabelle PG_SETTINGS e PG_LOCKS.

Ad esempio, non è possibile eseguire query su PG_STATS in una istruzione join con una tabella di Amazon Redshift a causa di funzioni non supportate.

Le viste e le tabelle di catalogo seguenti forniscono informazioni utili che possono essere unite alle informazioni nelle tabelle di Amazon Redshift. Alcune tabelle permettono solo l'accesso parziale a causa di restrizioni relative ai tipi di dati e alle funzioni. Quando si esegue una query su tabelle accessibili parzialmente, fare riferimento alle colonne o selezionarle con attenzione.

Le tabelle seguenti sono completamente accessibili e non contengono funzioni o tipi non supportati:

- pg_attribute
- pg_cast
- pg_depend
- pg_description
- pg_locks
- pg_opclass

Le tabelle seguenti sono parzialmente accessibili e contengono alcuni tipi e funzioni non supportati e colonne di testo troncate. I valori nelle colonne di testo sono troncati a valori varchar(256).

- pg_class
- pg_constraint
- pg_database
- pg_group
- pg_language
- pg_namespace
- pg_operator
- pg_proc
- pg_settings
- pg_statistic
- pg_tables
- pg_type
- pg_user
- pg_views

Le tabelle di catalogo non elencate non sono accessibili oppure è improbabile che siano utili per gli amministratori di Amazon Redshift. È possibile tuttavia eseguire una query su qualsiasi vista o tabella di catalogo se la query non richiede l'unione con una tabella di Amazon Redshift.

Puoi usare le colonne OID nelle tabelle di catalogo Postgres come colonne di unione. La condizione di unione `pg_database.oid = stv_tbl_perm.db_id`, ad esempio, corrisponde all'ID oggetto di database interno per ogni riga di PG_DATABASE con la colonna DB_ID visibile nella tabella STV_TBL_PERM. Le colonne OID sono chiavi primarie interne non visibili quando effettui la selezione nella tabella. Le viste di catalogo non hanno colonne OID.

Alcune funzioni di Amazon Redshift devono essere eseguite solo sui nodi di calcolo. Se una query fa riferimento a una tabella creata dall'utente, l'esecuzione di SQL avviene nei nodi di calcolo.

Una query che fa riferimento solo a tabelle di catalogo (tabelle con un prefisso PG, ad esempio PG_TABLE_DEF) o che non fa riferimento ad alcuna tabella viene eseguita esclusivamente sul nodo principale.

Se una query che usa una funzione di un nodo di calcolo non fa riferimento a una tabella definita dall'utente o a una tabella di sistema di Amazon Redshift, viene restituito il seguente errore.

```
[Amazon](500310) Invalid operation: One or more of the used functions must be applied
 on at least one user created table.
```

Le seguenti funzioni di Amazon Redshift sono funzioni applicate solo ai nodi di calcolo:

Funzioni di informazioni di sistema

- LISTAGG
- MEDIAN
- PERCENTILE_CONT
- PERCENTILE_DISC e APPROXIMATE PERCENTILE_DISC

Esempi di query di catalogo

Le query seguenti illustrano alcuni modi per eseguire query sulle tabelle di catalogo per ottenere informazioni utili su un database di Amazon Redshift.

Visualizza ID di tabella, database, schema e nome della tabella

La definizione di vista seguente unisce la tabella di sistema STV_TBL_PERM alle tabelle di catalogo di sistema PG_CLASS, PG_NAMESPACE e PG_DATABASE per restituire ID di tabella, nome del database, nome dello schema e nome della tabella.

```
create view tables_vw as
select distinct(id) table_id
,trim(datname)   db_name
,trim(nspname)   schema_name
,trim(relname)   table_name
from stv_tbl_perm
join pg_class on pg_class.oid = stv_tbl_perm.id
join pg_namespace on pg_namespace.oid = relnamespace
join pg_database on pg_database.oid = stv_tbl_perm.db_id;
```

L'esempio seguente restituisce le informazioni per la tabella con ID 117855.

```
select * from tables_vw where table_id = 117855;
```

```
table_id | db_name   | schema_name | table_name
```

```
---------+-----------+-------------+-----------
  117855 |       dev | public      | customer
```

Visualizzazione di un elenco del numero di colonne per la tabella di Amazon Redshift

La query seguente unisce alcune tabelle di catalogo per scoprire quante colonne contiene ogni tabella Amazon Redshift. I nomi delle tabelle di Amazon Redshift sono archiviati sia in PG_TABLES che in STV_TBL_PERM. Quando possibile, utilizzare PG_TABLES per restituire i nomi delle tabelle di Amazon Redshift.

Questa query non usa tabelle di Amazon Redshift.

```
select nspname, relname, max(attnum) as num_cols
from pg_attribute a, pg_namespace n, pg_class c
where n.oid = c.relnamespace and  a.attrelid = c.oid
and c.relname not like '%pkey'
and n.nspname not like 'pg%'
and n.nspname not like 'information%'
group by 1, 2
order by 1, 2;

nspname | relname  | num_cols
--------+----------+----------
public  | category |        4
public  | date     |        8
public  | event    |        6
public  | listing  |        8
public  | sales    |       10
public  | users    |       18
public  | venue    |        5
(7 rows)
```

Elenca gli schemi e le tabelle in un database

La query seguente unisce STV_TBL_PERM ad alcune tabelle PG per restituire un elenco delle tabelle nel database TICKIT con i relativi nomi di schemi (colonna NSPNAME). La query restituisce anche il numero totale di righe in ogni tabella. (Questa query è utile quando più schemi nel sistema hanno gli stessi nomi di tabella.)

```
select datname, nspname, relname, sum(rows) as rows
```

```
from pg_class, pg_namespace, pg_database, stv_tbl_perm
where pg_namespace.oid = relnamespace
and pg_class.oid = stv_tbl_perm.id
and pg_database.oid = stv_tbl_perm.db_id
and datname ='tickit'
group by datname, nspname, relname
order by datname, nspname, relname;

datname | nspname | relname  |  rows
--------+---------+----------+--------
tickit  | public  | category |     11
tickit  | public  | date     |    365
tickit  | public  | event    |   8798
tickit  | public  | listing  | 192497
tickit  | public  | sales    | 172456
tickit  | public  | users    |  49990
tickit  | public  | venue    |    202
(7 rows)
```

Elenca ID di tabella, tipi di dati, nomi di colonna e nomi di tabella

La query seguente elenca alcune informazioni su ogni tabella utente e sulle relative colonne: ID di tabella, nome di tabella, nomi delle colonne e tipo di dati per ogni colonna:

```
select distinct attrelid, rtrim(name), attname, typname
from pg_attribute a, pg_type t, stv_tbl_perm p
where t.oid=a.atttypid and a.attrelid=p.id
and a.attrelid between 100100 and 110000
and typname not in('oid','xid','tid','cid')
order by a.attrelid asc, typname, attname;

attrelid |  rtrim   |    attname     |  typname
---------+----------+----------------+-----------
  100133 | users    | likebroadway   | bool
  100133 | users    | likeclassical  | bool
  100133 | users    | likeconcerts   | bool
...
  100137 | venue    | venuestate     | bpchar
  100137 | venue    | venueid        | int2
  100137 | venue    | venueseats     | int4
  100137 | venue    | venuecity      | varchar
...
```

Conteggio del numero di blocchi di dati per ogni colonna in una tabella

La seguente query unisce la tabella STV_BLOCKLIST a PG_CLASS per restituire informazioni sullo storage per le colonne della tabella SALES.

```
select col, count(*)
from stv_blocklist s, pg_class p
where s.tbl=p.oid and relname='sales'
group by col
order by col;

col | count
----+-------
  0 |     4
  1 |     4
  2 |     4
  3 |     4
  4 |     4
  5 |     4
  6 |     4
  7 |     4
  8 |     4
  9 |     8
 10 |     4
 12 |     4
 13 |     8
(13 rows)
```

Informazioni di riferimento sulla configurazione

Argomenti

- spectrum_enable_pseudo_columns
- enable_spectrum_oid
- spectrum_query_maxerror
- statement_timeout
- stored_proc_log_min_messages
- timezone
- use_fips_ssl
- wlm_query_slot_count

Modifica della configurazione del server

È possibile modificare la configurazione del server nei seguenti modi:

- Usando un comando SET per sovrascrivere un'impostazione solo per la durata della sessione corrente.

  Ad esempio:

  ```
  set extra_float_digits to 2;
  ```

- Modificando le impostazioni del gruppo di parametri per il cluster. Le impostazioni del gruppo di parametri includono parametri aggiuntivi che è possibile configurare. Per ulteriori informazioni, consulta Gruppi di parametri di Amazon Redshift nella Guida alla gestione di Amazon Redshift.
- Usando il comando ALTER USER per impostare un parametro di configurazione su un nuovo valore per tutte le sessioni eseguite dall'utente specificato.

  ```
  ALTER USER username SET parameter { TO | = } { value | DEFAULT }
  ```

Usa il comando SHOW per visualizzare le impostazioni dei parametri correnti. Usa il comando SHOW ALL per visualizzare tutte le impostazioni che è possibile configurare con il comando SET.

```
SHOW ALL;

name                        | setting
----------------------------+--------------
```

```
analyze_threshold_percent | 10
datestyle                 | ISO, MDY
extra_float_digits        | 2
query_group               | default
search_path               | $user, public
statement_timeout         | 0
timezone                  | UTC
wlm_query_slot_count      | 1
```

> **Note**
>
> Tieni presente che i parametri di configurazione vengono applicati al database a cui sei connesso nel tuo data warehouse.

analyze_threshold_percent

Valori (valore predefinito in grassetto)

10, da 0 a 100.0

Description

Imposta la soglia per la percentuale di righe modificate per l'analisi di una tabella. Per ridurre il tempo di elaborazione e migliorare le prestazioni generali del sistema, Amazon Redshift ignora il comando ANALYZE per qualsiasi tabella che abbia una percentuale di righe modificate inferiore a quanto specificato da `analyze_threshold_percent`. Se, ad esempio, una tabella contiene 100.000.000 di righe e 9.000.000 di righe sono state modificate dall'ultima esecuzione di ANALYZE, per impostazione predefinita la tabella viene ignorata perché è stato modificato meno del 10% delle righe. Per analizzare le tabelle quando è stato modificato solo un piccolo numero di righe, imposta `analyze_threshold_percent` su un numero arbitrariamente piccolo. Ad esempio, se imposti `analyze_threshold_percent` su 0.01, una tabella con 100.000.000 di righe non verrà ignorata se sono state modificate almeno 10.000 righe. Per analizzare tutte le tabelle anche se non sono state modificate righe, imposta `analyze_threshold_percent` su 0.

È possibile modificare il parametro `analyze_threshold_percent` soltanto per la durata della sessione corrente utilizzando un comando SET. Il parametro non può essere modificato in un gruppo di parametri.

Esempio

```
set analyze_threshold_percent to 15;
set analyze_threshold_percent to 0.01;
set analyze_threshold_percent to 0;
```

cast_super_null_on_error

Valori (valore predefinito in grassetto)

on, off

Description

Specifica che quando si prova ad accedere a un membro inesistente di un oggetto o di un elemento di un array, Amazon Redshift restituisce un valore NULL se la query viene eseguita nella modalità permissiva di default.

datashare_break_glass_session_var

Valori (valore predefinito in grassetto)

Non c'è alcun valore predefinito. Il valore può essere qualsiasi stringa di caratteri generata da Amazon Redshift quando si verifica un'operazione non consigliata, come descritto di seguito.

Description

Applica un'autorizzazione che consente determinate operazioni che generalmente non sono consigliate per un AWS Data Exchange datashare.

In generale, si consiglia di non eliminare o modificare un AWS Data Exchange datashare utilizzando l'istruzione DROP DATASHARE o ALTER DATASHARE SET PUBLICACCESSIBLE. Per consentire l'eliminazione o la modifica di un AWS Data Exchange datashare per disattivare l'impostazione accessibile al pubblico, imposta la variabile su un valore monouso. `datashare_break_glass_session_var` Questo valore una tantum viene generato da Amazon Redshift e fornito in un messaggio di errore dopo il tentativo iniziale dell'operazione in questione.

Dopo aver impostato la variabile sul valore generato una tantum, eseguire nuovamente l'istruzione DROP DATASHARE o ALTER DATASHARE.

Per ulteriori informazioni, consulta Note per l'utilizzo di ALTER DATASHARE o Note per l'utilizzo di DROP DATASHARE.

Esempio

```
set datashare_break_glass_session_var to '620c871f890c49';
```

datestyle

Valori (valore predefinito in grassetto)

Specifica del formato (ISO, Postgres, SQL o German) e ordine di anno (Y)/mese (M)/giorno (D) (DMY, MDY, YMD).

- ISO: utilizza lo stile di data YYYY-MM-DD HH:MM:SS.
- Postgres: utilizza lo stile di data MM-DD HH:MM:SS YYYY.
- SQL: utilizza lo stile di data MM-DD-YYYY HH:MM:SS.
- Tedesco: utilizza lo stile di data DD-MM-YYYY HH:MM:SS.

Description

Imposta il formato di visualizzazione per i valori di data e ora, nonché le regole per interpretare i valori di input di data ambigui. La stringa contiene due parametri che possono essere modificati separatamente o insieme.

Esempio

```
show datestyle;
DateStyle
-----------
ISO, MDY
(1 row)
```

```
set datestyle to 'SQL,DMY';
```

default_geometry_encoding

Valori (valore predefinito in grassetto)

1, 2

Description

Una configurazione di sessione che specifica se le geometrie spaziali create durante una sessione sono codificate con un bounding box. Se `default_geometry_encoding` è `1`, le geometrie non vengono codificate con un bounding box. Se `default_geometry_encoding` è `2`, le geometrie vengono codificate con un bounding box. Per ulteriori informazioni sul supporto per i bounding box, consultare Riquadro di delimitazione.

describe_field_name_in_uppercase

Valori (valore predefinito in grassetto)

off (false), on (true)

Description

Specifica se i nomi delle colonne restituiti dalle istruzioni SELECT devono essere in maiuscolo o minuscolo. Se questo parametro è on, i nomi delle colonne vengono restituiti in lettere maiuscole. Se questo parametro è off, i nomi delle colonne vengono restituiti in lettere minuscole. Amazon Redshift archivia i nomi delle colonne in lettere minuscole indipendentemente dall'impostazione di `describe_field_name_in_uppercase`.

Esempio

```
set describe_field_name_in_uppercase to on;

show describe_field_name_in_uppercase;

DESCRIBE_FIELD_NAME_IN_UPPERCASE
--------------------------------
```

```
on
```

downcase_delimited_identifier

Valori (valore predefinito in grassetto)

on, off

Description

Questa configurazione sta per essere ritirata. Usare invece `enable_case_sensitive_identifier`.

Consente al super parser di leggere i campi JSON in maiuscolo o minuscolo. Consente inoltre il supporto delle query federate ai database PostgreSQL supportati con nomi di database, schema, tabella e colonna in maiuscolo e minuscolo. Per utilizzare identificatori con distinzione tra maiuscole e minuscole, impostare questo parametro su off.

Note per l'utilizzo

- Se utilizzi funzionalità di sicurezza a livello di riga o di mascheramento dinamico dei dati, consigliamo di impostare il valore `downcase_delimited_identifier` nel gruppo di parametri del cluster o del gruppo di lavoro. Ciò garantisce che `downcase_delimited_identifier` rimanga costante durante la creazione e l'associazione di una policy e quindi nell'esecuzione di una query su una relazione a cui è stato applicata una policy. Per ulteriori informazioni sulla sicurezza a livello di riga. consulta Sicurezza a livello di riga. Per informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.
- Quando disattivi `downcase_delimited_identifier` e crei una tabella, puoi impostare i nomi delle colonne con distinzione tra maiuscole e minuscole. Quando attivi `downcase_delimited_identifier` ed esegui query sulla tabella, i nomi delle colonne vengono riportati in lettere minuscole. Questo può produrre risultati di query diversi da quando `downcase_delimited_identifier` è disattivato. Considera il seguente esempio:

```
SET downcase_delimited_identifier TO off;
--Amazon Redshift preserves case for column names and other identifiers.

--Create a table with two columns that are identical except for the case.
CREATE TABLE t ("c" int, "C" int);
```

```
INSERT INTO t VALUES (1, 2);

SELECT * FROM t;

 c | C
---+---
 1 | 2
(1 row)

SET enable_downcase_delimited_identifier TO on;
--Amazon Redshift no longer preserves case for column names and other identifiers.

SELECT * FROM t;

 c | c
---+---
 1 | 1
(1 row)
```

- È consigliabile che gli utenti normali che eseguono query su tabelle con mascheramento dei dati dinamico o policy di sicurezza collegate a livello di riga abbiano l'impostazione predefinita downcase_delimited_identifier. Per ulteriori informazioni sulla sicurezza a livello di riga. consulta.Sicurezza a livello di riga. Per informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.

enable_case_sensitive_identifier

Valori (valore predefinito in grassetto)

true, false

Description

Un valore di configurazione che determina se gli identificatori dei nomi di database, tabelle e colonne fanno distinzione tra maiuscole e minuscole. Il formato maiuscolo/minuscolo degli identificatori del nome viene mantenuto quando racchiuso tra virgolette doppie. Quando si imposta `enable_case_sensitive_identifier` su `true`, il formato maiuscolo/minuscolo degli identificatori del nome viene mantenuto. Quando si imposta `enable_case_sensitive_identifier` su `false`, il formato maiuscolo/minuscolo degli identificatori del nome non viene mantenuto.

Il caso di un nome utente racchiuso tra virgolette doppie è sempre conservato indipendentemente dall'impostazione dell'opzione di configurazione `enable_case_sensitive_identifier`.

Esempi

Nell'esempio seguente viene illustrato come creare e utilizzare gli identificatori con distinzione tra maiuscole e minuscole per il nome di tabella e colonna.

```
-- To create and use case sensitive identifiers
SET enable_case_sensitive_identifier TO true;

-- Create tables and columns with case sensitive identifiers
CREATE TABLE "MixedCasedTable" ("MixedCasedColumn" int);

CREATE TABLE MixedCasedTable (MixedCasedColumn int);

-- Now query with case sensitive identifiers
SELECT "MixedCasedColumn" FROM "MixedCasedTable";

MixedCasedColumn
------------------
(0 rows)


SELECT MixedCasedColumn FROM MixedCasedTable;

mixedcasedcolumn
------------------
(0 rows)
```

Nell'esempio seguente viene illustrato quando non viene mantenuto il formato maiuscolo/minuscolo degli identificatori.

```
-- To not use case sensitive identifiers
RESET enable_case_sensitive_identifier;

-- Mixed case identifiers are lowercased
CREATE TABLE "MixedCasedTable2" ("MixedCasedColumn" int);

CREATE TABLE MixedCasedTable2 (MixedCasedColumn int);

ERROR:  Relation "mixedcasedtable2" already exists
```

```
SELECT "MixedCasedColumn" FROM "MixedCasedTable2";

 mixedcasedcolumn
------------------
(0 rows)


SELECT MixedCasedColumn FROM MixedCasedTable2;

 mixedcasedcolumn
------------------
(0 rows)
```

Note per l'utilizzo

- Se utilizzi l'aggiornamento automatico per le viste materializzate, consigliamo di impostare il valore `enable_case_sensitive_identifier` nel gruppo di parametri del cluster o del gruppo di lavoro. Ciò garantisce che `enable_case_sensitive_identifier` rimanga costante quando le viste materializzate vengono aggiornate. Per informazioni su come aggiornare automaticamente le viste materializzate, consulta Aggiornamento di una vista materializzata. Per informazioni relative all'impostazione dei valori di configurazione nei gruppi di parametri, consulta Gruppi di parametri di Amazon Redshift nella Guida alla gestione di Amazon Redshift.
- Se utilizzi funzionalità di sicurezza a livello di riga o di mascheramento dinamico dei dati, consigliamo di impostare il valore `enable_case_sensitive_identifier` nel gruppo di parametri del cluster o del gruppo di lavoro. Ciò garantisce che `enable_case_sensitive_identifier` rimanga costante durante la creazione e l'associazione di una policy e quindi nell'esecuzione di una query su una relazione a cui è stato applicata una policy. Per ulteriori informazioni sulla sicurezza a livello di riga. consulta Sicurezza a livello di riga. Per informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.
- Quando attivi `enable_case_sensitive_identifier` e crei una tabella, puoi impostare i nomi delle colonne con distinzione tra maiuscole e minuscole. Quando disattivi `enable_case_sensitive_identifier` ed esegui query sulla tabella, i nomi delle colonne vengono riportati in lettere minuscole. Questo può produrre risultati di query diversi da quando `enable_case_sensitive_identifier` è attivato. Considera il seguente esempio:

```
SET enable_case_sensitive_identifier TO on;
--Amazon Redshift preserves case for column names and other identifiers.
```

```
--Create a table with two columns that are identical except for the case.
CREATE TABLE t ("c" int, "C" int);

INSERT INTO t VALUES (1, 2);

SELECT * FROM t;

 c | C
---+---
 1 | 2
(1 row)

SET enable_case_sensitive_identifier TO off;
--Amazon Redshift no longer preserves case for column names and other identifiers.

SELECT * FROM t;

 c | c
---+---
 1 | 1
(1 row)
```

- È consigliabile che gli utenti normali che eseguono query su tabelle con mascheramento dei dati dinamico o policy di sicurezza collegate a livello di riga abbiano l'impostazione predefinita enable_case_sensitive_identifier. Per ulteriori informazioni sulla sicurezza a livello di riga. consulta Sicurezza a livello di riga. Per informazioni sul mascheramento dinamico dei dati, consulta Mascheramento dinamico dei dati.

enable_case_sensitive_super_attribute

Valori (valore predefinito in grassetto)

true, false

Description

Un valore di configurazione che determina se la navigazione delle strutture dei tipi di dati SUPER con nomi di attributi non delimitati fa distinzione tra maiuscole e minuscole. Quando imposti `enable_case_sensitive_super_attribute` su `true`, la navigazione nelle strutture di tipo SUPER con nomi di attributi non delimitati fa distinzione tra maiuscole e minuscole. Quando imposti

il valore su `false`, la navigazione nelle strutture di tipo SUPER con nomi di attributi non delimitati fa distinzione tra maiuscole e minuscole.

Quando racchiudi un nome di attributo tra virgolette doppie e imposti `enable_case_sensitive_identifier` su `true`, le maiuscole/minuscole sono sempre conservate, indipendentemente dall'impostazione dell'opzione di configurazione `enable_case_sensitive_super_attribute`.

`enable_case_sensitive_super_attribute` si applica solo alle colonne con il tipo di dati SUPER. Per tutte le altre colonne, considera invece l'utilizzo di `enable_case_sensitive_identifier`.

Per ulteriori informazioni sui tipi di dati SUPER, consulta Tipo SUPER e Importazione e query di dati semistrutturati in Amazon Redshift.

Esempi

L'esempio seguente mostra i risultati della selezione dei valori SUPER con `enable_case_sensitive_super_attribute` abilitato e disabilitato.

```
--Create a table with a SUPER column.
CREATE TABLE tbl (col SUPER);

--Insert values.
INSERT INTO tbl VALUES (json_parse('{
 "A": "A", "a": "a"
}'));

SET enable_case_sensitive_super_attribute TO ON;

SELECT col.A FROM tbl;
  a
-----
 "A"
(1 row)

SELECT col.a FROM tbl;
  a
-----
 "a"
(1 row)
```

```
SET enable_case_sensitive_super_attribute TO OFF;

SELECT col.A FROM tbl;
  a
-----
 "a"
(1 row)

SELECT col.a FROM tbl;
  a
-----
 "a"
(1 row)
```

Note per l'utilizzo

- Le viste e le viste materializzate seguono il valore di `enable_case_sensitive_super_attribute` al momento della loro creazione. Le viste con associazione tardiva, le procedure archiviate e le funzioni definite dall'utente seguono il valore di `enable_case_sensitive_super_attribute` al momento della query.
- Se utilizzi l'aggiornamento automatico per le viste materializzate, consigliamo di impostare `enable_case_sensitive_identifier value` nel gruppo di parametri del cluster o del gruppo di lavoro. Ciò garantisce che `enable_case_sensitive_identifier` rimanga costante quando le viste materializzate vengono aggiornate. Per informazioni su come aggiornare automaticamente le viste materializzate, consulta Aggiornamento di una vista materializzata. Per informazioni relative all'impostazione dei valori di configurazione nei gruppi di parametri, consulta Gruppi di parametri di Amazon Redshift nella Guida alla gestione di Amazon Redshift.
- Il nome della colonna nei risultati dell'istruzione è sempre in minuscolo, indipendentemente dal valore di `enable_case_sensitive_super_attribute`. Per fare in modo che anche il nome della colonna faccia distinzione tra maiuscole e minuscole, abilita `enable_case_sensitive_identifier`.
- È consigliabile che gli utenti normali che eseguono query su tabelle con policy di sicurezza collegate a livello di riga abbiano l'impostazione predefinita `enable_case_sensitive_identifier`. Per ulteriori informazioni sulla sicurezza a livello di riga. consulta.Sicurezza a livello di riga.

enable_numeric_rounding

Valori (valore predefinito in grassetto)

attivato (true), disattivato (false)

Description

Specifica se utilizzare l'arrotondamento numerico. Se `enable_numeric_rounding` è `on`, Amazon Redshift arrotonda i valori NUMERIC quando li trasmette ad altri tipi numerici, come INTEGER o DECIMAL. Se `enable_numeric_rounding` è `off`, Amazon Redshift tronca i valori NUMERIC quando li trasmette ad altri tipi numerici. Per ulteriori informazioni sui tipi numerici, consulta Tipi numerici.

Esempio

```
--Create a table and insert the numeric value 1.5 into it.
CREATE TABLE t (a numeric(10, 2));

INSERT INTO t VALUES (1.5);

SET enable_numeric_rounding to ON;
--Amazon Redshift now rounds NUMERIC values when casting to other numeric types.

SELECT a::int FROM t;

 a
---
 2
(1 row)


SELECT a::decimal(10, 0) FROM t;

 a
---
 2
(1 row)


 SELECT a::decimal(10, 5) FROM t;
```

```
    a
---------
 1.50000
(1 row)


SET enable_numeric_rounding to OFF;
--Amazon Redshift now truncates NUMERIC values when casting to other numeric types.

SELECT a::int FROM t;

 a
---
 1
(1 row)


SELECT a::decimal(10, 0) FROM t;

 a
---
 1
(1 row)


SELECT a::decimal(10, 5) FROM t;

    a
---------
 1.50000
(1 row)
```

enable_result_cache_for_session

Valori (valore predefinito in grassetto)

on (true), off (false)

Description

Specifica se memorizzare nella cache i risultati delle query. Se `enable_result_cache_for_session` è on, Amazon Redshift verifica la presenza di una copia

valida e memorizzata nella cache dei risultati della query quando viene inviata una query. Se viene trovata una corrispondenza nella cache dei risultati, Amazon Redshift usa i risultati memorizzati nella cache e non esegue la query. Se `enable_result_cache_for_session` è `off`, Amazon Redshift ignora la cache dei risultati ed esegue tutte le query che vengono inviate.

Esempio

```
SET enable_result_cache_for_session TO off;
--Amazon Redshift now ignores the results cache
```

enable_vacuum_boost

Valori (valore predefinito in grassetto)

false, true

Description

Specifica se abilitare l'opzione vacuum boost per tutti i comandi VACUUM eseguiti in una sessione Se `enable_vacuum_boost` è `true`, Amazon Redshift esegue tutti i comandi VACUUM nella sessione applicando l'opzione BOOST. Se `enable_vacuum_boost` è `false`, Amazon Redshift non viene eseguito con l'opzione BOOST per impostazione predefinita. Per ulteriori informazioni sull'opzione BOOST, consultare VACUUM.

error_on_nondeterministic_update

Valori (valore predefinito in grassetto)

false, true

Description

Specifica se le query UPDATE con più corrispondenze per riga generano un errore.

Esempio

```
SET error_on_nondeterministic_update TO true;
```

```
CREATE TABLE t1(x1 int, y1 int);

CREATE TABLE t2(x2 int, y2 int);

INSERT INTO t1 VALUES (1,10), (2,20), (3,30);

INSERT INTO t2 VALUES (2,40), (2,50);

UPDATE t1 SET y1=y2 FROM t2 WHERE x1=x2;

ERROR: Found multiple matches to update the same tuple.
```

extra_float_digits

Valori (valore predefinito in grassetto)

0, da -15 a 2

Description

Imposta il numero di cifre visualizzato per i valori in virgola mobile, inclusi float4 e float8. Il valore viene aggiunto al numero standard di cifre (FLT_DIG o DBL_DIG, come appropriato). Il valore può essere impostato al massimo su 2, per includere cifre parzialmente significative. Ciò è particolarmente utile per l'emissione di dati FLOAT che devono essere ripristinati esattamente. In alternativa, è possibile impostare un valore negativo per eliminare le cifre indesiderate.

Esempio

L'esempio seguente imposta `extra_float_digits` su `-2`. Innanzitutto, mostra l'impostazione corrente dei parametri.

```
show all;
      name                      | setting
--------------------------+----------------
analyze_threshold_percent | 10
datestyle                 | ISO, MDY
extra_float_digits        | 2
query_group               | default
search_path               | $user, public
statement_timeout         | 0
timezone                  | UTC
```

```
wlm_query_slot_count      | 1
```

Quindi, imposta il nuovo valore su `-2`.

```
set extra_float_digits to -2;
```

Infine, mostra l'impostazione aggiornata dei parametri.

```
show all;
      name                      | setting
--------------------------+----------------
analyze_threshold_percent | 10
datestyle                 | ISO, MDY
extra_float_digits        | -2
query_group               | default
search_path               | $user, public
statement_timeout         | 0
timezone                  | UTC
wlm_query_slot_count      | 1
```

interval_forbid_composite_literals

Valori (valore predefinito in grassetto)

falso, vero

Description

Una configurazione di sessione che modifica il valore di un intervallo che contiene sia le parti DA ANNO A MESE che da GIORNO A SECONDO.

In caso `interval_forbid_composite_literals` `true` affermativo, viene restituito un errore se viene rilevato un intervallo con le parti ANNO A MESE e GIORNO A SECONDO. Ad esempio, il codice SQL seguente contiene un INTERVALLO DA UN GIORNO AL SECONDO con le parti DA ANNO A MESE e DA GIORNO A SECONDO.

```
SELECT INTERVAL '1 year 1 day' DAY TO SECOND;
ERROR:  Interval Day To Second literal cannot contain year-month parts. Disable the GUC
 interval_forbid_composite_literals to suppress this error and silently discard the
 year-month part.
```

In caso `interval_forbid_composite_literals` affermativo`false`, Amazon Redshift elimina un errore e tronca la parte YEAR-TO-MONTH da un valore INTERVAL DAY TO SECOND. Ad esempio, il seguente codice SQL contiene un INTERVALLO DAY TO SECOND con entrambe le parti ANNO - MESE e GIORNO - SECONDO.

```
SET interval_forbid_composite_literals to "false";
SELECT INTERVAL '1 year 1 day' DAY TO SECOND;


 intervald2s
------------------------------
1 days 0 hours 0 mins 0.0 secs
```

json_serialization_enable

Valori (valore predefinito in grassetto)

false, true

Description

Una configurazione di sessione che modifica il comportamento di serializzazione JSON dei dati formattati ORC, JSON, Ion e Parquet. Se `json_serialization_enable` è `true`, tutte le raccolte di livello superiore vengono serializzate automaticamente in JSON e restituite come VARCHAR(65535). Le colonne non complesse non sono interessate o serializzate. Quando le colonne di raccolta sono serializzate come VARCHAR(65535), non è possibile accedere direttamente ai relativi sottocampi nidificati come parte della sintassi della query (ad esempio, nella clausola filter). Se `json_serialization_enable` è `false`, le raccolte di livello superiore non vengono serializzate in JSON. Per ulteriori informazioni sulla serializzazione JSON nidificata, consultare Serializzazione di JSON nidificato complesso.

json_serialization_parse_nested_strings

Valori (valore predefinito in grassetto)

false, true

Description

Una configurazione di sessione che modifica il comportamento di serializzazione JSON dei dati formattati ORC, JSON, Ion e Parquet. Quando sia `json_serialization_parse_nested_strings` che `json_serialization_enable` sono true, i valori stringa memorizzati in tipi complessi (ad esempio mappe, strutture o array) vengono analizzati e scritti in linea direttamente nel risultato se sono JSON validi. Se `json_serialization_parse_nested_strings` è false, le stringhe all'interno di tipi complessi nidificati vengono serializzate come stringhe JSON con escape. Per ulteriori informazioni, consultare Serializzazione di tipi complessi contenenti stringhe JSON.

max_concurrency_scaling_clusters

Valori (valore predefinito in grassetto)

1, da 0 a 10

Description

Imposta il numero massimo di cluster di dimensionamento simultaneo consentiti quando è abilitato il dimensionamento simultaneo. Aumenta questo valore se è richiesto un maggiore dimensionamento della simultaneità. Diminuisci questo valore per ridurre l'utilizzo dei cluster di dimensionamento della simultaneità e i costi di fatturazione risultanti.

Il numero massimo di cluster di dimensionamento simultaneo è una quota regolabile. Per informazioni, consulta Quote di Amazon Redshift nella Guida alla gestione di Amazon Redshift.

max_cursor_result_set_size

Valori (valore predefinito in grassetto)

0 (viene usato come predefinito il valore massimo) - 14400000 MB

Description

Il parametro `max_cursor_result_set_size` non è più in uso. Per ulteriori informazioni sulle dimensioni del set di risultati del cursore, consultare Vincoli del cursore.

mv_enable_aqmv_for_session

Valori (valore predefinito in grassetto)

true, false

Description

Specifica se Amazon Redshift può eseguire la riscrittura automatica delle query delle viste materializzate a livello di sessione.

navigate_super_null_on_error

Valori (valore predefinito in grassetto)

on, off

Description

Specifica che quando si prova a passare a un membro inesistente di un oggetto o di un elemento di un array, Amazon Redshift restituisce un valore NULL se la query viene eseguita nella modalità permissiva di default.

parse_super_null_on_error

Valori (valore predefinito in grassetto)

off, on

Description

Specifica che quando Amazon Redshift prova ad analizzare un membro inesistente di un oggetto o di un elemento di un array, Amazon Redshift restituisce un valore NULL se la query viene eseguita nella modalità rigorosa.

pg_federation_repeatable_read

Valori (valore predefinito in grassetto)

true, false

Description

Specifica il livello di isolamento delle transazioni di query federate per i risultati del database PostgreSQL.

- Quando `pg_federation_repeatable_read` è true, le transazioni federate vengono elaborate con la semantica del livello di isolamento REPETABLE READ. Questa è l'impostazione predefinita.
- Quando `pg_federation_repeatable_read` è false, le transazioni federate vengono elaborate con la semantica del livello di isolamento READ COMMITTED.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Considerazioni quando si accede ai dati federati con Amazon Redshift.
- Gestione delle operazioni di scrittura simultanee.

Esempi

Il comando seguente imposta `pg_federation_repeatable_read` su on per una sessione. Il comando SHOW mostra il valore del valore impostato.

```
set pg_federation_repeatable_read to on;
```

```
show pg_federation_repeatable_read;

pg_federation_repeatable_read
-----------------------------
on
```

query_group

Valori (valore predefinito in grassetto)

Nessun valore predefinito. Il valore può essere qualsiasi stringa di caratteri.

Description

Applica un'etichetta definita dall'utente a un gruppo di query eseguite durante la stessa sessione. Questa etichetta viene acquisita nei log delle query. Può essere usata per limitare i risultati dalle tabelle STL_QUERY e STV_INFLIGHT e dalla vista SVL_QLOG. Puoi ad esempio applicare un'etichetta distinta per ogni query eseguita per identificare in modo univoco le query senza doverne esaminare gli ID.

Questo parametro non è presente nel file di configurazione del server e deve essere impostato in fase di runtime con un comando SET. Anche se è possibile usare come etichetta una stringa di caratteri lunga, l'etichetta viene troncata a 30 caratteri nella colonna LABEL della tabella STL_QUERY e della vista SVL_QLOG (e a 15 caratteri in STV_INFLIGHT).

Nell'esempio seguente il valore di query_group è impostato su **`Monday`** e quindi vengono eseguite diverse query con tale etichetta.

```
set query_group to 'Monday';
SET
select * from category limit 1;
...
...
select query, pid, substring, elapsed, label
from svl_qlog where label ='Monday'
order by query;

query | pid  |        substring                    |  elapsed  | label
------+------+-------------------------------------+-----------+--------
789   | 6084 | select * from category limit 1;     | 65468     | Monday
790   | 6084 | select query, trim(label) from ... | 1260327   | Monday
791   | 6084 | select * from svl_qlog where ..     | 2293547   | Monday
792   | 6084 | select count(*) from bigsales;      | 108235617 | Monday
...
```

search_path

Valori (valore predefinito in grassetto)

'$user', public, nomi_schemi

Elenco di nomi di schemi esistenti separati da virgola. Se è presente '$user', lo schema con lo stesso nome di `SESSION_USER` viene sostituito, altrimenti viene ignorato.

Description

Specifica l'ordine con cui vengono eseguite le ricerche negli schemi quando si fa riferimento a un oggetto (ad esempio, una tabella o una funzione) usando un nome semplice senza componente dello schema.

- I percorsi di ricerca non sono supportati con gli schemi esterni e le tabelle esterne. Le tabelle esterne devono essere qualificate in modo esplicito da uno schema esterno.
- Quando vengono creati oggetti senza uno schema di destinazione specifico, gli oggetti vengono inseriti nel primo schema elencato nel percorso di ricerca. Se il percorso di ricerca è vuoto, il sistema restituisce un errore.
- Quando sono presenti oggetti con nomi identici in schemi diversi, viene usato l'oggetto trovato per primo nel percorso di ricerca.
- È possibile fare riferimento a un oggetto non incluso in alcuno schema nel percorso di ricerca solo specificando lo schema che lo contiene con un nome completo (puntato).
- Nello schema del catalogo di sistema, pg_catalog, la ricerca al suo interno viene sempre eseguita. Se tale schema è menzionato nel percorso, la ricerca viene eseguita in base all'ordine specificato. In caso contrario, la ricerca viene eseguita prima che in qualsiasi altro elemento del percorso.
- Se lo schema di tabella temporanea della sessione corrente, pg_temp_nnn, esiste, la ricerca al suo interno viene sempre eseguita. Può essere elencato in modo esplicito nel percorso usando l'alias pg_temp. Se lo schema non è elencato nel percorso, la ricerca al suo interno viene eseguita come prima cosa (anche prima della ricerca in pg_catalog). Tuttavia, nello schema temporaneo viene eseguita solo la ricerca dei nomi di relazioni (tabelle, viste). Non viene eseguita la ricerca dei nomi di funzioni.

Esempio

L'esempio seguente crea lo schema ENTERPRISE e imposta search_path sul nuovo schema.

```
create schema enterprise;
set search_path to enterprise;
show search_path;

 search_path
-------------
 enterprise
(1 row)
```

L'esempio seguente aggiunge lo schema ENTERPRISE al percorso search_path predefinito.

```
set search_path to '$user', public, enterprise;
show search_path;

         search_path
-----------------------------
 "$user", public, enterprise
(1 row)
```

L'esempio seguente aggiunge la tabella FRONTIER allo schema ENTERPRISE.

```
create table enterprise.frontier (c1 int);
```

Quando viene creata la tabella PUBLIC.FRONTIER nello stesso database e l'utente non specifica il nome dello schema in una query, PUBLIC.FRONTIER ha la precedenza su ENTERPRISE.FRONTIER.

```
create table public.frontier(c1 int);
insert into enterprise.frontier values(1);
select * from frontier;

frontier
----
(0 rows)

select * from enterprise.frontier;

c1
----
1
(1 row)
```

spectrum_enable_pseudo_columns

Valori (valore predefinito in grassetto)

true, false

Description

Puoi disabilitare la creazione di pseudocolonne per una sessione impostando il parametro di configurazione `spectrum_enable_pseudo_columns` su `false`.

Esempio

Il seguente comando disabilita la creazione di pseudocolonne per una sessione.

```
set spectrum_enable_pseudo_columns to false;
```

enable_spectrum_oid

Valori (valore predefinito in grassetto)

true, false

Description

È anche possibile disabilitare la pseudocolonna `$spectrum_oid` impostando il parametro di configurazione `enable_spectrum_oid` su `false`.

Esempio

Il seguente comando disabilita la pseudocolonna `$spectrum_oid` impostando il parametro di configurazione `enable_spectrum_oid` su `false`.

```
set enable_spectrum_oid to false;
```

spectrum_query_maxerror

Valori (valore predefinito in grassetto)

-1, numero intero

Description

Scopri come specificare un numero intero per indicare il numero massimo di errori da accettare prima di annullare la query. Un valore negativo disattiva la gestione del numero massimo di dati di errore. I risultati sono registrati in SVL_SPECTRUM_SCAN_ERROR.

Esempio

L'esempio seguente presuppone i dati ORC che contengono caratteri eccedenti e caratteri non validi. La definizione di colonna per `my_string` specifica una lunghezza di 3 caratteri. Di seguito vengono riportati dati di esempio per questo esempio.

```
my_string
---------------
abcdef
gh
ab
```

I seguenti comandi impostano il numero massimo di errori su 1 ed eseguono la query.

```
set spectrum_query_maxerror to 1;
SELECT my_string FROM orc_data;
```

La query si interrompe e i risultati vengono registrati in SVL_SPECTRUM_SCAN_ERROR.

statement_timeout

Valori (valore predefinito in grassetto)

0 (disattiva la limitazione), x millisecondi

Description

Interrompe le istruzioni che impiegano un tempo superiore al numero di millisecondi specificato.

Il valore `statement_timeout` è il tempo massimo che una query può essere eseguita prima che Amazon Redshift la termini. Questo tempo include la pianificazione, l'accodamento nella gestione del carico di lavoro (WLM) e il tempo di esecuzione. Confronta questo tempo con il timeout WLM (max_execution_time) e QMR (query_execution_time), che includono solo i tempi di esecuzione.

Se viene specificato anche un timeout di gestione dei carichi di lavoro (WLM, Workload Management) (max_execution_time) come parte di una configurazione WLM, viene usato il valore inferiore tra statement_timeout e max_execution_time. Per ulteriori informazioni, consultare Timeout WLM.

Esempio

Poiché la query seguente impiega più di 1 millisecondo, si verifica il timeout e la query viene annullata.

```
set statement_timeout = 1;

select * from listing where listid>5000;
ERROR:  Query (150) canceled on user's request
```

stored_proc_log_min_messages

Valori (valore predefinito in grassetto)

LOG, INFO, NOTICE, WARNING, EXCEPTION

Description

Specifica il livello di registrazione minimo dei messaggi delle procedure archiviate generati. Vengono registrati messaggi al livello specificato o a un livello superiore. Il valore predefinito è LOG (tutti i messaggi sono registrati). L'ordine dei livelli di log dal più alto al più basso è:

1. EXCEPTION
2. WARNING
3. NOTICE
4. INFO
5. LOG

Ad esempio, se si specifica un valore NOTICE, i messaggi vengono registrati solo per NOTICE, WARNING ed EXCEPTION.

timezone

Valori (valore predefinito in grassetto)

UTC, fuso orario

Sintassi

```
SET timezone { TO | = } [ time_zone | DEFAULT ]

SET time zone [ time_zone | DEFAULT ]
```

Description

Imposta il fuso orario per la sessione corrente. Il fuso orario può essere l'offset dall'ora UTC (Universal Coordinated Time) o il nome di un fuso orario.

> **Note**
>
> Non è possibile impostare il parametro di configurazione `timezone` usando un gruppo di parametri del cluster. Il fuso orario può essere impostato solo per la sessione corrente usando un comando SET. Per impostare il fuso orario per tutte le sessioni eseguite da un utente di database specifico, usa il comando ALTER USER. ALTER USER … SET TIMEZONE modifica il fuso orario per le sessioni successive e non per quella corrente.

Quando imposti il fuso orario usando il comando `SET timezone` (una sola parola) con TO o =, puoi specificare time_zone come nome di fuso orario, offset in formato in stile POSIX oppure offset in formato ISO-8601, come illustrato di seguito.

```
SET timezone { TO | = } time_zone
```

Quando il fuso orario viene impostato usando il comando SET time zone senza TO o =, è possibile specificare time_zone usando INTERVAL oltre che un nome di fuso orario, un offset in formato in stile POSIX oppure un offset in formato ISO-8601, come illustrato di seguito.

```
SET time zone time_zone
```

Formati di fuso orario

Amazon Redshift supporta i formati di fuso orario seguenti:

- Nome di fuso orario
- INTERVAL
- Specifica del fuso orario in stile POSIX
- Offset ISO-8601

Poiché le abbreviazioni dei fusi orari, come PST o PDT, sono definite come offset fisso rispetto a UTC e non includono le regole relative all'ora legale, non sono supportare dal comando SET.

Per ulteriori informazioni sui formati di fuso orario, consultare quanto segue.

Nome fuso orario: il nome completo del fuso orario, ad esempio America/New_York. I nomi completi dei fusi orari possono includere le regole relative all'ora legale.

Di seguito sono elencati alcuni esempio di nomi di fuso orario:

- Etc/Greenwich
- America/New_York
- CST6CDT
- GB

> **Note**
>
> Molti nomi di fuso orario, ad esempio EST, MST, NZ e UCT, sono anche abbreviazioni.

Per visualizzare un elenco di nomi di fuso orario validi, esegui il comando seguente.

```
select pg_timezone_names();
```

INTERVAL: un offset da UTC. Ad esempio, PST è -8:00 o -8 ore.

Di seguito sono elencati alcuni esempio di offset di fuso orario INTERVAL:

- -8:00
- -8 ore
- 30 minutes

Formato in stile POSIX: una specifica di fuso orario nel formato STDoffset o STDoffsetDST, dove STD è un'abbreviazione di fuso orario, offset è l'offset numerico in ore a ovest di UTC e DST è un'abbreviazione facoltativa di fuso orario con ora legale. L'ora legale viene considerata come un'ora avanti rispetto all'offset specificato.

I formati di fuso orario in stile POSIX utilizzano offset positivi a ovest di Greenwich, contrariamente alla convenzione ISO-8601, che utilizza offset positivi a est di Greenwich.

Di seguito sono riportati alcuni esempi di fusi orari in stile POSIX:

- PST8
- PST8PDT
- EST5
- EST5EDT

> **Note**
>
> Amazon Redshift non convalida le specifiche di fuso orario in stile POSIX, di conseguenza è possibile impostare il fuso orario su un valore non valido. Ad esempio, il comando seguente non restituisce un errore, anche se si imposta il fuso orario su un valore non valido.
>
> ```
> set timezone to ‘xxx36’;
> ```

Offset ISO-8601: l'offset da UTC nel modulo `±[hh]:[mm]`.

Di seguito vengono illustrati alcuni esempi di offset ISO-8601:

- -8:00
- +7:30

Esempi

L'esempio seguente imposta il fuso orario per la sessione corrente su New York.

```
set timezone = 'America/New_York';
```

L'esempio seguente imposta il fuso orario per la sessione corrente su UTC–8 (PST).

```
set timezone to '-8:00';
```

L'esempio seguente usa INTERVAL per impostare il fuso orario su PST.

```
set timezone interval '-8 hours'
```

L'esempio seguente reimposta il fuso orario per la sessione corrente sul fuso orario predefinito di sistema (UTC).

```
set timezone to default;
```

Per impostare il fuso orario per un utente di database, usa un'istruzione ALTER USER … SET. L'esempio seguente imposta il fuso orario per dbuser su New York. Il nuovo valore rimane valido per l'utente per tutte le sessioni successive.

```
ALTER USER dbuser SET timezone to 'America/New_York';
```

use_fips_ssl

Valori (valore predefinito in grassetto)

true, false

Description

Un valore di gruppo di parametri che specifica se viene utilizzata la modalità SSL conforme a FIPS. In caso affermativo, `use_fips_ssl` viene utilizzata la modalità SSL conforme a `true` FIPS. In caso affermativo`false`, `use_fips_ssl` la modalità SSL conforme a FIPS non viene utilizzata. Per

ulteriori informazioni, consulta Configurazione delle opzioni di sicurezza per le connessioni nella Amazon Redshift Management Guide.

Per configurare i parametri per un cluster con provisioning di Amazon Redshift, consulta Informazioni sui gruppi di parametri nella Amazon Redshift Management Guide. Per configurare i parametri per Redshift Serverless, consulta Configurazione di una connessione SSL conforme a FIPS ad Amazon Redshift Serverless nella Amazon Redshift Management Guide e/o nel Redshift Serverless API Reference. CreateWorkgroupUpdateWorkgroup

wlm_query_slot_count

Valori (valore predefinito in grassetto)

1, da 1 a 50 (il valore non può essere superiore al numero di slot disponibili (livello di simultaneità) per la classe di servizio)

Description

Imposta il numero di slot di query che una query userà.

La gestione del flusso di lavoro (WLM) riserva degli slot in una classe di servizio in base al livello di simultaneità impostato per la coda. Ad esempio, se il livello di simultaneità è impostato su 5, la classe di servizio ha 5 slot. La gestione dei carichi di lavoro alloca la memoria disponibile per una classe di servizio a ogni slot. Per ulteriori informazioni, consultare Implementazione della gestione del carico di lavoro.

> **Note**
>
> Se il valore di wlm_query_slot_count è superiore al numero di slot disponibili (livello di simultaneità) per la classe di servizio, la query ha esito negativo. Se si verifica un errore, diminuisci wlm_query_slot_count scegliendo un valore permesso.

Per le operazioni in cui le prestazioni sono influenzate dalla quantità di memoria allocata, ad esempio il vacuum, aumentando il valore di wlm_query_slot_count è possibile migliorare le prestazioni. In particolare, per i comandi di vacuum, esamina il record corrispondente nella vista SVV_VACUUM_SUMMARY. Se sono presenti valori elevati (intoro a 100 o superiori) per sort_partitions e merge_increments nella vista SVV_VACUUM_SUMMARY, aumenta il valore per wlm_query_slot_count alla successiva esecuzione di Vacuum sulla tabella.

Aumentando il valore di wlm_query_slot_count si limita il numero di query simultanee che possono essere eseguite. Si supponga, ad esempio, che la classe di servizio abbia un livello di simultaneità pari a 5 e che il valore di wlm_query_slot_count sia impostato su 3. Durante l'esecuzione di una query nella sessione con il valore di wlm_query_slot_count impostato su 3 possono essere eseguite al massimo altre 2 query simultanee nella stessa classe di servizio. Le query successive attendono in coda che le query in esecuzione simultaneamente vengano completate e gli slot si liberino.

Esempi

Usa il comando SET per impostare il valore di wlm_query_slot_count per la durata della sessione corrente.

```
set wlm_query_slot_count to 3;
```

Cronologia dei documenti

> **Note**
>
> Per una descrizione delle nuove funzionalità di Amazon Redshift, consulta What's new.

La tabella seguente descrive le importanti modifiche alla documentazione apportate alla Amazon Redshift Database Developer Guide dopo maggio 2018. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi abbonarti a un feed RSS.

Versione API: 2012-12-01

Per un elenco delle modifiche apportate alla Guida alla gestione di Amazon Redshift, consulta Cronologia dei documenti della guida alla gestione di Amazon Redshift.

Per ulteriori informazioni sulle nuove caratteristiche, incluso un elenco delle correzioni e dei numeri di versione dei cluster associati per ogni release, consultare cronologia della versione del cluster.

Modifica	Descrizione	Data
Support per geometrie spaziali 3D e 4D e nuove funzioni spaziali	Ora è possibile utilizzare funzioni spaziali aggiuntive e il supporto geometrico 3D e 4D viene aggiunto ad alcune funzioni.	19 agosto 2021
Support per la codifica a compressione delle colonne per l'ottimizzazione automatica delle tabelle	È possibile specificare l'opzione ENCODE AUTO per la tabella per gestire automaticamente la codifica di compressione per tutte le colonne della tabella.	3 agosto 2021
Support per più istruzioni SQL o un'istruzione SQL con	Ora puoi eseguire più istruzion i SQL o un'istruzione con	28 luglio 2021

parametri che utilizzano l'API Amazon Redshift Data	parametri con l'API Amazon Redshift Data.	
Support per confronto senza distinzione tra maiuscole e minuscole con sostituzioni a livello di colonna	Ora è possibile utilizzare la clausola COLLATE all'interno di un'istruzione CREATE DATABASE per specificare il confronto predefinito.	24 giugno 2021
Support per la condivisione dei dati tra account	Ora puoi condividere i dati tra Account AWS.	30 aprile 2021
Support per query gerarchiche di dati con CTE ricorrente	Ora è possibile utilizzare un'espressione di tabella comune ricorrente (CTE) nel proprio SQL.	29 aprile 2021
Support per le query tra database	Ora è possibile eseguire query sui dati tra i database in un cluster.	10 marzo 2021
Support per il controllo granulare degli accessi sui comandi COPY e UNLOAD	Ora puoi concedere il privilegio di eseguire i comandi COPY e UNLOAD a utenti e gruppi specifici nel cluster Amazon Redshift per creare policy di controllo degli accessi più dettagliati.	12 gennaio 2021
Supporto per dati JSON e semistrutturati nativi	Ora è possibile definire il tipo di dati SUPER.	9 dicembre 2020
Supporto per query federate su MySQL	Ora è possibile scrivere una query federata su un motore MySQL supportato.	9 dicembre 2020
Supporto per la condivisione dei dati	Ora è possibile condividere i dati tra i cluster Amazon Redshift.	9 dicembre 2020

Supporto per l'ottimizzazione automatica delle tabelle	È ora possibile definire le chiavi di distribuzione e ordinamento automatiche.	9 dicembre 2020
Supporto per l'ML di Amazon Redshift	Ora è possibile creare, addestrare e implementare modelli di Machine Learning (ML).	8 dicembre 2020
Supporto per l'aggiornamento automatico e la riscrittura delle query delle viste materializzate	Ora puoi mantenere le viste materializzate up-to-date con l'aggiornamento automatico e le prestazioni delle query possono essere migliorate con la riscrittura automatica.	11 novembre 2020
Supporto per i tipi di dati TIME e TIMETZ	È ora possibile creare tabelle con i tipi di dati TIME e TIMETZ. Il tipo di dati TIME memorizza l'ora del giorno senza informazioni sul fuso orario mentre TIMETZ memorizza l'ora del giorno includendo le informazioni sul fuso orario	11 novembre 2020
Supporto per funzioni Lambda definite dall'utente e tokenizzazione	Ora è possibile scrivere funzioni Lambda definite dall'utente per abilitare la tokenizzazione esterna dei dati.	26 ottobre 2020
Supporto per la modifica di una codifica di colonne di tabella	È ora possibile modificare la codifica delle colonne di una tabella.	20 ottobre 2020

Supporto per l'esecuzione di query tra database	Amazon Redshift adesso può eseguire query tra database in un cluster.	15 ottobre 2020
Support per HyperLogLog schizzi	Amazon Redshift ora può archiviare ed elaborare. HyperLogLogSketches	2 ottobre 2020
Supporto per Apache Hudi e Delta Lake	Miglioramenti alla creazione di tabelle esterne per Redshift Spectrum.	24 settembre 2020
Supporto per miglioramenti all'esecuzione di query su dati spaziali	I miglioramenti includono il caricamento di uno shapefile e diverse nuove funzioni SQL spaziali.	15 settembre 2020
Le viste materializzate supportano tabelle esterne	È possibile creare viste materializzate in Amazon Redshift che fanno riferimento a origini dati esterne.	19 giugno 2020
Supporto per la scrittura su una tabella esterna	È possibile scrivere su tabelle esterne eseguendo CREATE EXTERNAL TABLE AS SELECT per scrivere in una nuova tabella esterna o INSERT INTO per inserire dati in una tabella esterna esistente.	8 giugno 2020
Supporto per i controlli di archiviazione per gli schemi	Aggiornamenti ai comandi e alle viste che gestiscono i controlli di archiviazione per gli schemi.	2 giugno 2020

Supporto per la disponibilità generale delle query federate	Informazioni aggiornate sull'interrogazione dei dati con query federate.	16 aprile 2020
Supporto per funzioni spaziali aggiuntive	Aggiunte descrizioni di funzioni spaziali aggiuntive.	2 aprile 2020
Supporto per la disponibilità generale delle viste materializzate	Le viste materializzate sono generalmente disponibili a partire dalla versione del cluster 1.0.13059.	19 febbraio 2020
Supporto per i privilegi a livello di colonna	I privilegi a livello di colonna sono disponibili a partire dalla versione del cluster 1.0.13059.	19 febbraio 2020
ALTER TABLE	È possibile utilizzare un comando ALTER TABLE con la clausola ALTER DISTSTYLE ALL per modificare lo stile di distribuzione di una tabella.	11 febbraio 2020
Supporto per query federate	È stata aggiornata la guida per descrivere la query federata con un CREATE EXTERNAL SCHEMA aggiornato.	3 dicembre 2019
Supporto per l'esportazione di data lake	È stata aggiornata la guida per descrivere i nuovi parametri del comando UNLOAD.	3 dicembre 2019
Supporto dei dati spaziali	Aggiornamento della guida per descrivere il supporto dei dati spaziali.	21 novembre 2019

Supporto per la nuova console	Aggiornata la guida per descrivere la nuova console di Amazon Redshift.	11 novembre 2019
Supporto dell'ordinamento automatico delle tabelle	Amazon Redshift è in grado di ordinare automaticamente i dati delle tabelle.	7 novembre 2019
Supporto dell'opzione VACUUM BOOST	È possibile utilizzare l'opzione BOOST durante la pulizia delle tabelle.	7 novembre 2019
Supporto per colonne IDENTITY predefinite	Puoi creare tabelle con colonne IDENTITY predefinite.	19 settembre 2019
Supporto per la codifica di compressione AZ64	Puoi codificare alcune colonne con codifica di compressione AZ64.	19 settembre 2019
Supporto per la priorità delle query	Puoi impostare la priorità delle query di una coda WLM automatico.	22 agosto 2019
Supporto perAWS Lake Formation	È possibile utilizzare un catalogo di dati di Lake Formation con Amazon Redshift Spectrum.	8 agosto 2019
COMPUPDATE PRESET	È possibile utilizzare un comando COPY con COMPUPDATE PRESET per consentire ad Amazon Redshift di scegliere la codifica di compressione.	13 giugno 2019

ALTER COLUMN	È possibile utilizzare un comando ALTER TABLE con ALTER COLUMN per aumentare le dimensioni di una colonna VARCHAR.	22 maggio 2019
Supporto per le procedure archiviate	È possibile definire le stored procedure PL/pgSQL in Amazon Redshift.	24 aprile 2019
Supporto per la configurazione di una gestione automatica del carico di lavoro (WLM, Workload Management)	È possibile abilitare Amazon Redshift affinché venga eseguito con il WLM automatic o.	24 aprile 2019
UNLOAD to Zstandard	È possibile utilizzare il comando UNLOAD per applicare la compressione Zstandard ai file di testo e CSV (valori delimitati da virgole) scaricati in Amazon S3.	3 aprile 2019
Dimensionamento simultaneo	Quando il dimensionamento simultaneo è abilitato, Amazon Redshift aggiunge automatic amente ulteriore capacità del cluster quando necessario per elaborare un aumento delle query di lettura simultanee.	21 marzo 2019
UNLOAD to CSV	Puoi utilizzare il comando UNLOAD per scaricare in un file formattato come testo CSV (valori delimitati da virgole).	13 marzo 2019

Stile di distribuzione AUTO	Per abilitare la distribuzione automatica, è possibile specificare lo stile di distribuzione AUTO con un'istruzione CREATE TABLE. Quando si abilita la distribuzione automatica, Amazon Redshift assegna uno stile di distribuzione ottimale basato sulla dimensione dei dati della tabella. La modifica della distribuzione avviene in background, in pochi secondi.	23 gennaio 2019
Il comando COPY da Parquet supporta SMALLINT	Ora il comando COPY supporta il caricamento da file con formattazione Parquet in colonne che utilizzano il tipo di dati SMALLINT. Per ulteriori informazioni, consultare COPY da formati di dati a colonna.	2 gennaio 2019
DROP EXTERNAL DATABASE	È possibile eliminare un database esterno includendo la clausola DROP EXTERNAL DATABASE con un comando DROP SCHEMA.	3 dicembre 2018
UNLOAD tra regioni	È possibile eseguire l'UNLOAD in un bucket Amazon S3 in un'altra regione AWS specificando il parametro REGION.	31 ottobre 2018

Eliminazione vacuum automatica	Amazon Redshift esegue automaticamente una operazione VACUUM DELETE in background, per cui non sarà quasi mai necessario eseguire un vacuum DELETE ONLY. Amazon Redshift pianificherà l'esecuzione di VACUUM DELETE durante i periodi di carico ridotto e sospenderà l'operazione durante i periodi di alto carico.	31 ottobre 2018
Distribuzione automatica	Se non si specifica uno stile di distribuzione con l'istruzione CREATE TABLE, Amazon Redshift assegna uno stile di distribuzione ottimale basato sui dati della tabella. La modifica della distribuzione avviene in background, in pochi secondi.	31 ottobre 2018
Controllo granulare degli accessi per il AWS Glue Data Catalog	È ora possibile specificare i livelli di accesso ai dati archiviati nel AWS Glue Data Catalog.	15 ottobre 2018
UNLOAD con tipi di dati	È possibile specificare l'opzione MANIFEST VERBOSE con un comando UNLOAD per aggiungere i metadati al file manifest, compresi i nomi e i tipi di dati di colonne, dimensioni di file e conteggi di righe.	10 ottobre 2018

Aggiungere più partizioni tramite una singola istruzione ALTER TABLE	Per le tabelle esterne Redshift Spectrum, è possibile combinare più clausole PARTITION in una singola istruzione ALTER TABLE ADD. Per ulteriori informazioni, consultare Esempi per alterare la tabella esterna.	10 ottobre 2018
UNLOAD con intestazione	È possibile specificare l'opzione HEADER con un comando UNLOAD per aggiungere una riga di intestazione contenente nomi di colonne nella parte superiore di ciascun file di output.	19 settembre 2018
Nuove viste e tabella di sistema	Aggiunta la documentazione SVL_S3Retries, SVL_USER_INFO e STL_DISK_FULL_DIAG.	31 agosto 2018
Supporto per i dati nidificati con Amazon Redshift Spectrum	È ora possibile eseguire una query per i dati nidificati archiviati nelle tabelle Amazon Redshift Spectrum. Per ulteriori informazioni sull'esecuzione di query sui dati annidati, consultare Tutorial: esecuzione di una query sui dati annidati con Amazon Redshift Spectrum.	8 agosto 2018

SQA attivata per impostazione predefinita	L'accelerazione di query brevi (SQA) è ora abilitata per impostazione predefinita per tutti i nuovi cluster. SQA utilizza il machine learning per fornire performance migliori, risultati più veloci e una maggiore prevedibilità dei tempi di esecuzione delle query. Per ulteriori informazioni su Transfer Acceleration, consultare Accelerazione di query brevi.	8 agosto 2018
Amazon Redshift Advisor	Ora puoi ottenere consigli personalizzati su come migliorare le prestazioni dei cluster e ridurre i costi operativi mediante Amazon Redshift Advisor. Per ulteriori informazioni, consultare Amazon Redshift Advisor.	26 luglio 2018
Riferimento alias immediato	Ora puoi fare riferimento a un espressione con alias subito dopo averla definita. Per ulteriori informazioni, consultare SELECT List.	18 luglio 2018
Specifica del tipo di compressione quando si crea una tabella esterna	Ora puoi specificare il tipo di compressione alla creazione di una tabella esterna con Amazon Redshift Spectrum. Per ulteriori informazioni, consultare Creazione di tabelle esterne.	27 giugno 2018

PG_LAST_UNLOAD_ID	Aggiunta la documentazione per una nuova funzione di informazione del sistema: PG_LAST_UNLOAD_ID. Per ulteriori informazioni, consultare PG_LAST_UNLOAD_ID.	27 giugno 2018
ALTER TABLE RENAME COLUMN	ALTER TABLE ora supporta la ridenominazione delle colonne per le tabelle esterne. Per ulteriori informazioni, consultare Esempi per alterare la tabella esterna.	7 giugno 2018

Aggiornamenti precedenti

Nella tabella seguente sono descritte le modifiche importanti introdotte in ogni versione della Guida per gli sviluppatori di database di Amazon Redshift prima di giugno 2018.

Modifica	Descrizione	Data della modifica
Il comando COPY da Parquet include SMALLINT	Ora il comando COPY supporta il caricamento da file con formattazione Parquet in colonne che utilizzan o il tipo di dati SMALLINT. Per ulteriori informazioni, consultare COPY da formati di dati a colonna	2 gennaio 2019
COPY per formati a colonne	COPY ora supporta il caricamento da file in Amazon S3 che utilizzano i formati di dati a colonne Parquet e ORC. Per ulteriori informazioni, consultare COPY da formati di dati a colonna	17 maggio 2018
Tempo di esecuzione massimo dinamico per SQA	Per impostazione predefinita, WLM ora assegna dinamicamente un valore per il tempo di esecuzione massimo SQA in base all'analisi del carico di lavoro del	17 maggio 2018

Modifica	Descrizione	Data della modifica
	cluster. Per ulteriori informazioni, consultare Tempo di esecuzione massimo per query brevi.	
Nuova colonna in STL_LOAD_COMMITS	La tabella di sistema STL_LOAD_COMMITS include una nuova colonna, `file_format` .	10 maggio 2018
Nuove colonne in STL_HASHJOIN e altre tabelle di log di sistema	La tabella di sistema STL_HASHJOIN ha tre nuove colonne, `hash_segment` , `hash_step` e `checksum`. Inoltre, un `checksum` è stato aggiunto a STL_MERGEJOIN, STL_NESTLOOP, STL_HASH, STL_SCAN, STL_SORT, STL_LIMIT e STL_PROJECT.	17 maggio 2018
Nuove colonne in STL_AGGR	La tabella di sistema STL_AGGR ha due nuove colonne, `resizes` e `flushable` .	19 aprile 2018
Nuove opzioni per le funzioni REGEX	Per le funzioni REGEXP_INSTR e REGEXP_SUBSTR, ora è possibile specificare quale occorrenza di una corrispondenza utilizzare e se eseguire una corrispondenza con distinzione tra maiuscole e minuscole . REGEXP_INSTR consente inoltre di specificare se restituire la posizione del primo carattere della corrispondenza o la posizione del primo carattere dopo la fine della corrispondenza.	22 marzo 2018
Nuove colonne nelle tabelle di sistema	Le colonne tombstonedblocks, tossedblocks e batched_by columns sono state aggiunte alla tabella di sistema STL_COMMIT_STATS. La colonna localslice column è stata aggiunta alla vista di sistema STV_SLICES.	22 marzo 2018

Modifica	Descrizione	Data della modifica
Aggiunta ed eliminazione di colonne in tabelle esterne	ALTER TABLE ora supporta ADD COLUMN e DROP COLUMN per le tabelle esterne di Amazon Redshift Spectrum.	22 marzo 2018
Nuove regioni AWS per Redshift Spectrum	Redshift Spectrum è ora disponibile nelle regioni di Mumbai e di San Paolo. Per un elenco delle regioni supportate, consultare Regioni di Amazon Redshift Spectrum.	22 marzo 2018
Limite delle tabelle è passato a 20.000	Il numero massimo di tabelle è ora 20.000 per i tipi di nodi dei cluster 8xlarge. Il limite per i tipi di nodi large e xlarge node è 9.900. Per ulteriori informazioni, consultare Limiti e quote.	13 marzo 2018
Supporto di Redshift Spectrum per JSON e Ion	Con Redshift Spectrum, è possibile fare riferimento a file con dati scalari nei formati di dati JSON o Ion. Per ulteriori informazioni, consultare CREATE EXTERNAL TABLE.	26 febbraio 2018
Concatenazione di ruoli IAM per Redshift Spectrum	È possibile concatenare ruoli AWS Identity and Access Management (IAM) in modo che il cluster possa assumere altri ruoli non associati al cluster, tra cui ruoli appartenenti a un altro account AWS. Per ulteriori informazioni, consultare Concatenazione di ruoli IAM per Amazon Redshift Spectrum.	1 febbraio 2018
ADD PARTITION supporta IF NOT EXISTS	La clausola ADD PARTITION per ALTER TABLE ora supporta un'opzione IF NOT EXISTS. Per ulteriori informazioni, consultare ALTER TABLE.	11 gennaio 2018
Dati DATE per tabelle esterne	Ora le tabelle esterne di Redshift Spectrum supportano il tipo di dati DATE. Per ulteriori informazioni, consultare CREATE EXTERNAL TABLE.	11 gennaio 2018

Modifica	Descrizione	Data della modifica
Nuove regioni AWS per Redshift Spectrum	Redshift Spectrum è ora disponibile nelle regioni di Singapore, Sydney, Seoul e Francoforte. Per un elenco delle regioni AWS supportate, consultare Regioni di Amazon Redshift Spectrum.	16 novembre 2017
Accelerazione di query brevi nella gestione dei carichi di lavoro (WLM) di Amazon Redshift	L'accelerazione di query brevi (SQA) rende prioritarie le query a esecuzione breve rispetto a quelle a esecuzione prolungata. SQA esegue le query a esecuzione breve in uno spazio dedicato, di modo che le query SQA non siano costrette ad attendere nelle code dietro le query più lunghe. Con SQA, l'esecuzione delle query brevi è più rapida e gli utenti vedono prima i risultati. Per ulteriori informazioni, consultare Utilizzo dall'accelerazione di query brevi.	16 novembre 2017
WLM riassegna le query sottoposte a hop	Anziché annullare e riavviare una query sottoposta a hop, la gestione dei carichi di lavoro (WLM) di Amazon Redshift ora riassegna le query idonee a una nuova coda. Quando WLM riassegna una query, la sposta nella nuova coda e continua l'esecuzione, risparmiando quindi tempo e risorse di sistema. Le query sottoposte a hop che non possono essere riassegnate vengono riavviate o annullate. Per ulteriori informazioni, consultare Hop della coda di query WLM.	16 novembre 2017
Accesso ai log di sistema per gli utenti	Nella maggior parte delle tabelle di log di sistema visibili agli utenti, per impostazione predefinita le righe generate da un altro utente sono invisibili a un utente standard. Per consentire a un utente standard di vedere tutte le righe delle tabelle visibili agli utenti, incluse le righe generate da un altro utente, eseguire ALTER USER o CREA UTENTE e impostare il parametro SYSLOG ACCESS su UNRESTRICTED;	16 novembre 2017

Modifica	Descrizione	Data della modifica
Caching dei risultati	Con Caching dei risultati, quando si esegue una query, Amazon Redshift memorizza nella cache i risultati. Quando si esegue di nuovo la query, Amazon Redshift cerca una copia valida del risultato della query nella cache. Se viene trovata una corrispondenza nella cache dei risultati, Amazon Redshift usa i risultati memorizzati nella cache e non esegue la query. Per impostazione predefinita, il caching dei risultati è abilitato. Per disabilitare il caching dei risultati, imposta il parametro di configurazione enable_result_cache_for_session su `off`.	16 novembre 2017
Funzioni per metadati di colonne	PG_GET_COLS e PG_GET_LATE_BINDING_VIEW_COLS restituiscono i metadati di colonne per tabelle, viste e viste con associazione tardiva di Amazon Redshift.	16 novembre 2017
Salto di code WLM per CTAS	La gestione dei carichi di lavoro (WLM) di Amazon Redshift ora supporta l'hop delle code di query per le istruzioni CREATE TABLE AS (CTAS) nonché query di sola lettura, come le istruzioni SELECT. Per ulteriori informazioni, consultare Hop della coda di query WLM.	19 ottobre 2017
File manifest di Amazon Redshift Spectrum	Quando si crea una tabella esterna di Redshift Spectrum, è possibile specificare un file manifest che elenca le posizioni dei file di dati in Amazon S3. Per ulteriori informazioni, consultare CREATE EXTERNAL TABLE.	19 ottobre 2017
Nuove regioni AWS per Amazon Redshift Spectrum	Redshift Spectrum è ora disponibile nelle regioni Europa (Irlanda) e Asia Pacifico (Tokyo). Per un elenco delle regioni AWS supportate, consultare Considerazioni su Amazon Redshift Spectrum.	19 ottobre 2017

Modifica	Descrizione	Data della modifica
Formati file aggiunti per Amazon Redshift Spectrum	Ora puoi creare tabelle esterne di Redshift Spectrum basate sui formati di file di dati Regex, OpenCSV e Avro. Per ulteriori informazioni, consultare CREATE EXTERNAL TABLE.	5 ottobre 2017
Pseudocolonne per tabelle esterne di Amazon Redshift Spectrum	È possibile selezionare le pseudocolonne `$path` e `$size` in una tabella esterna di Redshift Spectrum per visualizzare la posizione e la dimensione dei file di dati a cui si fa riferimento in Amazon S3. Per ulteriori informazioni, consultare Pseudocolonne.	5 ottobre 2017
Funzioni di convalida JSON	Puoi utilizzare le funzioni IS_VALID_JSON e IS_VALID_JSON_ARRAY per verificare la formattaz ione JSON. Le altre funzioni JSON hanno ora un argomento `null_if_invalid` facoltativo.	5 ottobre 2017
LISTAGG DISTINCT	Puoi utilizzare la clausola DISTINCT con la funzione di aggregazione LISTAGG e la funzione finestra LISTAGG per eliminare valori duplicati dall'espressione specificata prima di eseguire la concatenazione.	5 ottobre 2017
Visualizzazione dei nomi delle colonne in maiuscole	Per visualizzare i nomi delle colonne in maiuscolo nei risultati SELECT, puoi impostare il parametro di configurazione describe_field_name_in_uppercase su `true`.	5 ottobre 2017
Ignorare le righe di intestazione nelle tabelle esterne	Puoi impostare la proprietà `skip.header.line.c ount` nel comando CREATE EXTERNAL TABLE per ignorare le righe di intestazione all'inizio dei file di dati Redshift Spectrum.	5 ottobre 2017

Modifica	Descrizione	Data della modifica
Analizza conteggio righe	Le regole di monitoraggio di query WLM utilizzano il parametro scan_row_count metric per restituire il numero di righe in una fase SCAN. Il conteggio delle righe è il numero totale di righe generate prima di aver applicato filtri alle righe contrassegnate per l'elimina zione (righe fantasma) e prima dell'applicazione di filtri di query definiti dall'utente. Per ulteriori informazioni, consultare Metriche per il monitoraggio delle query per Amazon Redshift.	21 settembre 2017
Funzioni SQL definite dall'utente	Una funzione SQL scalare definita dall'utente (UDF) integra una clausola SQL SELECT che viene eseguita quando la funzione viene chiamata e restituisce un singolo valore. Per ulteriori informazioni, consultare Creazione di una funzione definita dall'utente SQL scalare.	31 agosto 2017
Viste con associazione tardiva	Una vista con associazione tardiva non è vincolata agli oggetti del database sottostante, come le tabelle e le funzioni definite dall'utente. Di conseguenza, non esiste alcuna dipendenza tra la vista e gli oggetti a cui fa riferimento. Puoi creare una vista anche se gli oggetti di riferimento non esistono. Poiché non esiste alcuna dipendenza, è possibile rimuovere o modificar e un oggetto di riferimento senza influire sulla vista. Amazon Redshift non controlla le dipendenze finché non viene interrogata la vista. Per creare una vista con associazione tardiva, specifica la clausola WITH NO SCHEMA BINDING con l'istruzione CREATE VIEW. Per ulteriori informazioni, consultare CREATE VIEW.	31 agosto 2017
Funzione OCTET_LENGTH	OCTET_LENGTH restituisce la lunghezza della stringa specificata come numero di byte.	18 agosto 2017

Modifica	Descrizione	Data della modifica
Tipi di file ORC e Grok supportati	Amazon Redshift Spectrum ora supporta i formati di dati ORC e Grok per i file di dati di Redshift Spectrum. Per ulteriori informazioni, consultare Creazione di file di dati per le query in Amazon Redshift Spectrum.	18 agosto 2017
RegexSerDe ora supportato	Amazon Redshift Spectrum ora supporta RegexSerD e il formato dei dati. Per ulteriori informazioni, consulta Creazione di file di dati per le query in Amazon Redshift Spectrum.	19 luglio 2017
Nuove colonne aggiunte a SVV_TABLES e SVV_COLUMNS	Le colonne `domain_name` e `remarks` sono state aggiunte a SVV_COLUMNS. Una colonna di commenti è stata aggiunta a SVV_TABLES.	19 luglio 2017
Viste di sistema SVV_TABLES e SVV_COLUMNS	Le viste di sistema SVV_TABLES e SVV_COLUMNS forniscono informazioni sulle colonne e altri dettagli per le tabelle e le viste locali ed esterne.	7 luglio 2017
VPC non più necessario per Amazon Redshift Spectrum con metastore Hive di Amazon EMR	Redshift Spectrum ha rimosso l'obbligo di avere il cluster Amazon Redshift e il cluster Amazon EMR nello stesso VPC e nella stessa sottorete quando si utilizza un metastore Hive di Amazon EMR. Per ulteriori informazioni, consultare Utilizzo di cataloghi esterni in Amazon Redshift Spectrum.	7 luglio 2017
UNLOAD per dimensioni di file più piccole	Per impostazione predefinita, UNLOAD crea più file in Amazon S3 con una dimensione massima di 6,2 GB. Per creare file più piccoli, specifica MAXFILESI ZE con il comando UNLOAD. Puoi specificare una dimensione di file massima tra 5 MB e 6,2 GB. Per ulteriori informazioni, consultare UNLOAD.	7 luglio 2017

Modifica	Descrizione	Data della modifica
TABLE PROPERTIES	Ora puoi impostare il parametro TABLE PROPERTIES numRows per CREATE EXTERNAL TABLE o ALTER TABLE allo scopo di aggiornare le statistiche di tabelle e riflettere il numero di righe nella tabella.	6 giugno 2017
ANALYZE PREDICATE COLUMNS	Per risparmiare tempo e risorse del cluster, puoi scegliere di analizzare solo le colonne che potrebber o essere utilizzate come predicati. Quando esegui ANALYZE con la clausola PREDICATE COLUMNS, l'operazione di analisi include solo le colonne che sono state utilizzate in una condizione di join, una condizion e di filtro o una clausola group by oppure utilizzat e come chiave di ordinamento o chiave di distribuz ione. Per ulteriori informazioni, consultare Analisi delle tabelle.	25 maggio 2017
Policy IAM per Amazon Redshift Spectrum	Per concedere l'accesso a un bucket Amazon S3 solo mediante Redshift Spectrum, è possibile includere una condizione che consenta l'accesso per l'agente utente `AWS Redshift/Spectrum` . Per ulteriori informazioni, consultare Policy IAM per Amazon Redshift Spectrum.	25 maggio 2017
Scansione ricorsiva di Amazon Redshift Spectrum	Redshift Spectrum ora esegue la scansione dei file nelle sottocartelle nonché nella cartella specificata in Amazon S3. Per ulteriori informazioni, consultare Creazione di tabelle esterne per Redshift Spectrum.	25 maggio 2017

Modifica	Descrizione	Data della modifica
Regole di monitoraggio delle query	Grazie alle regole di monitoraggio delle query WLM, è possibile definire i limiti delle prestazioni basati sui parametri per le code WLM e specificare l'azione da intraprendere quando una query oltrepassa tali limiti, ovvero registrazione, hop o interruzione. Le regole di monitoraggio delle query vengono definite come parte della configurazione della gestione del carico di lavoro (WLM). Per ulteriori informazioni, consultare Regole di monitoraggio delle query WLM.	21 Aprile 2017
Amazon Redshift Spectrum	Con Redshift Spectrum, è possibile eseguire una query e recuperare in modo efficace dati dai file in Amazon S3 senza doverli caricare nelle tabelle. Le query di Redshift Spectrum vengono eseguite molto rapidamente su set di dati di grandi dimensioni in quanto Redshift Spectrum esegue la scansione di file di dati direttamente in Amazon S3. L'elaborazione viene eseguita principalmente nel livello di Amazon Redshift Spectrum e la maggior parte dei dati rimane in Amazon S3. Più cluster possono eseguire simultane amente query sullo stesso set di dati in Amazon S3 senza la necessità di effettuare copie dei dati per ogni cluster. Per ulteriori informazioni, consultare Esecuzion e di query sui dati esterni utilizzando Amazon Redshift Spectrum	19 aprile 2017

Modifica	Descrizione	Data della modifica
Nuove tabelle di sistema per supportare Redshift Spectrum	Le seguenti nuove viste di sistema sono state aggiunte per supportare Redshift Spectrum: • SVL_S3QUERY • SVL_S3QUERY_SUMMARY • SVV_EXTERNAL_COLUMNS • SVV_EXTERNAL_DATABASES • SVV_EXTERNAL_PARTITIONS • SVV_EXTERNAL_TABLES • PG_EXTERNAL_SCHEMA	19 aprile 2017
Funzione di aggregazione APPROXIMATE PERCENTILE_DISC	La funzione di aggregazione APPROXIMATE PERCENTILE_DISC è ora disponibile.	4 Aprile 2017
Crittografia lato server con KMS	Ora è possibile scaricare dati in Amazon S3 utilizzando la crittografia lato server con una chiave AWS Key Management Service (SSE-KMS). Inoltre, COPY carica ora in modo trasparente i file di dati crittografati con KMS da Amazon S3. Per ulteriori informazioni, consultare UNLOAD.	9 febbraio 2017

Modifica	Descrizione	Data della modifica
Nuova sintassi di autorizzazione	Ora puoi utilizzare i parametri IAM_ROLE, MASTER_SYMMETRIC_KEY, ACCESS_KEY_ID, SECRET_ACCESS_KEY e SESSION_TOKEN pe fornire informazioni di autorizzazione e accesso per i comandi COPY, UNLOAD e CREATE LIBRARY La nuova sintassi di autorizzazione fornisce un'alternativa più flessibile di fornire un singolo argomento di stringa al parametro CREDENTIALS. Per ulteriori informazi oni, consultare Parametri di autorizzazione.	9 febbraio 2017
Aumento del limite di schemi	Ora puoi creare fino a 9.900 schemi per cluster. Per ulteriori informazioni, consultare CREATE SCHEMA.	9 febbraio 2017
Codifica della tabelle predefinita	Ora CREATE TABLE e ALTER TABLE assegnano la codifica di compressione LZO alla maggior parte delle nuove colonne. Per impostazione predefinita, alle colonne designate come chiavi di ordinamento e come tipi di dati BOOLEAN, REAL o DOUBLE PRECISION e alle tabelle temporanee viene assegnata la codifica RAW. Per ulteriori informazioni, consultare ENCODE.	6 febbraio 2017
Codifica di compressione ZSTD	Amazon Redshift supporta ora la codifica di compressi one di colonne ZSTD.	19 gennaio 2017
Funzioni di aggregazione PERCENTIL E_CONT e MEDIAN	PERCENTILE_CONT e MEDIAN sono ora disponibi li come funzioni di aggregazione e come funzioni finestra.	19 gennaio 2017

Modifica	Descrizione	Data della modifica
Registrazione utente UDF	Puoi usare il modulo di logging Python per creare messaggi di errore e avviso definiti dall'utente nelle funzioni definite dall'utente. In seguito all'esecuzione della query, puoi eseguire una query sulla vista di sistema SVL_UDF_LOG per recuperare i messaggi registrati. Per ulteriori informazioni sui messaggi definiti dall'utente, consultare Logging di errori e avvisi in funzioni definite dall'utente	8 dicembre 2016
Stima della riduzione con ANALYZE COMPRESSION	Il comando ANALYZE COMPRESSION ora indica una stima della percentuale di riduzione dello spazio su disco per ogni colonna. Per ulteriori informazioni, consultare ANALYZE COMPRESSION.	10 Novembre 2016
Limiti di connessioni	Ora puoi impostare un limite per il numero di connessioni di database simultanee che un utente può aprire. Puoi inoltre limitare il numero di connessioni simultanee per un database. Per ulteriori informazioni, consultare CREA UTENTE e CREATE DATABASE.	10 Novembre 2016
Miglioramento dell'ordine delle chiavi di ordinamento con COPY	Il comando COPY ora aggiunge automaticamente nuove righe alla regione ordinata della tabella quando carichi i dati nell'ordine delle chiave di ordinamento. Per informazioni sulle condizioni di abilitazione di questo miglioramento, consultare Caricamento dei dati nell'ordine delle chiavi di ordinamento	10 Novembre 2016
CTAS con compressione	CREATE TABLE AS (CTAS) ora assegna automaticamente le codifiche di compressione alle nuove tabelle in funzione del tipo di dati della colonna. Per ulteriori informazioni, consultare Ereditarietà di attributi di colonna e tabella.	28 Ottobre 2016

Modifica	Descrizione	Data della modifica
Tipo di dati timestamp con fuso orario	Amazon Redshift ora supporta un tipo di dati timestamp con fuso orario (TIMESTAMPTZ). Alcune nuove funzioni sono state inoltre aggiunte per supportare il nuovo tipo di dati. Per ulteriori informazioni, consultare Funzioni di data e ora.	29 settembre 2016
Soglia di analisi	Per ridurre i tempi di elaborazione e migliorare le prestazioni globali del sistema per le operazioni ANALYZE, Amazon Redshift non esegue l'analisi di una tabella se la percentuale di righe che sono state modificate dall'ultima esecuzione del comando ANALYZE è inferiore alla soglia di analisi specificata dal parametro analyze_threshold_percent. Per impostazione predefinita, `analyze_threshold_percent` è 10.	9 agosto 2016
Nuova tabella di sistema STL_RESTARTED_SESSIONS	Quando Amazon Redshift riavvia una sessione, STL_RESTARTED_SESSIONS registra il nuovo ID di processo (PID) e quello vecchio.	9 agosto 2016
Aggiornata la documentazione relativa alle funzioni di data e ora	Aggiunta di un riepilogo delle funzioni con collegamenti a Funzioni di data e ora e aggiornamento dei riferimenti delle funzioni per scopi di coerenza.	24 giugno 2016
Nuove colonne in STL_CONNECTION_LOG	La tabella di sistema STL_CONNECTION_LOG include due nuove colonne per tenere traccia delle connessioni SSL. Se si caricano regolarmente i log di audit in una tabella di Amazon Redshift, sarà necessario aggiungere alla tabella di destinazione le seguenti nuove colonne: sslcompression e sslexpansion.	5 maggio 2016

Modifica	Descrizione	Data della modifica
Password hash MD5	Puoi specificare la password per un comando CREA UTENTE o ALTER USER fornendo la stringa hash MD5 della password e il nome utente.	21 aprile 2016
Nuova colonna in STV_TBL_PERM	La colonna `backup` nella vista di sistema STV_TBL_PERM indica se la tabella è inclusa negli snapshot del cluster. Per ulteriori informazioni, consultare BACKUP.	21 aprile 2016
Tabelle senza backup	Per le tabelle che non contengono dati critici, come le tabelle di gestione temporanea, è possibile specifica re BACKUP NO nell'istruzione CREATE TABLE o CREATE TABLE AS per impedire ad Amazon Redshift di includere la tabella negli snapshot automatici o manuali. L'utilizzo delle tabelle senza backup riduce il tempo di elaborazione durante la creazione di snapshot e il ripristino da snapshot nonché lo spazio di archiviazione in Amazon S3.	7 Aprile 2016
Soglia di eliminazi one VACUUM	Per impostazione predefinita, il comando VACUUM consente ora di recuperare spazio in modo tale che almeno il 95 percento delle restanti righe non viene contrassegnato per l'eliminazione. Di conseguenza, il comando VACUUM richiede in genere molto meno tempo per la fase di eliminazione rispetto al recupero del 100% delle righe eliminate. È possibile modificare la soglia predefinita per una singola tabella includendo il parametro TO soglia PERCENT quando si esegue il comando VACUUM.	7 Aprile 2016
Tabella di sistema SVV_TRANS ACTIONS	La vista di sistema SVV_TRANSACTIONS registra le informazioni sulle transazioni che attualmente mantengono blocchi sulle tabelle nel database.	7 Aprile 2016

Modifica	Descrizione	Data della modifica
Utilizzo di ruoli IAM per accedere ad altre risorse AWS	Per spostare i dati tra il cluster e un'altra risorsa AWS, ad esempio Amazon S3, Amazon DynamoDB, Amazon EMR o Amazon EC2, il cluster deve disporre dell'autorizzazione di accedere alla risorsa e di eseguire le operazioni necessarie. Come alternativa più sicura per fornire una coppia di chiavi di accesso con i comandi COPY, UNLOAD o CREATE LIBRARY, puoi ora specificare un ruolo IAM che il cluster utilizza per l'autenticazione e l'autorizzazione. Per ulteriori informazioni, consultare Controllo degli accessi basato sui ruoli.	29 marzo 2016
Soglia di ordinamento VACUUM	Il comando VACUUM ora ignora la fase di ordinamento per ogni tabella dove più del 95% delle righe della tabella sono già ordinate. È possibile modificare la soglia di ordinamento predefinita per una singola tabella includendo il parametro TO threshold PERCENT quando si esegue il comando VACUUM.	17 marzo 2016
Nuove colonne in STL_CONNECTION_LOG	La tabella di sistema STL_CONNECTION_LOG include tre nuove colonne. Se i log di verifica vengono caricati regolarmente in una tabella di Amazon Redshift, sarà necessario aggiungere alla tabella di destinazione le seguenti nuove colonne: sslversion, sslcipher e mtu.	17 marzo 2016
UNLOAD con compressione bzip2	Ora hai la possibilità di utilizzare il comando UNLOAD per eseguire lo scaricamento mediante la compressione bzip2.	8 febbraio 2016

Modifica	Descrizione	Data della modifica
ALTER TABLE APPEND	ALTER TABLE APPEND aggiunge righe a una tabella di destinazione spostando i dati da una tabella di origine esistente. ALTER TABLE APPEND è in genere molto più veloce di un'operazione CREATE TABLE AS o INSERT INTO simile in quanto i dati vengono spostati, non duplicati.	8 febbraio 2016
Hop della coda di query WLM	Se WLM annulla una query di sola lettura, come un'istruzione SELECT, a causa di un timeout WLM, WLM tenta di instradare la query alla coda corrispon dente successiva. Per ulteriori informazioni, consultare Hop della coda di query WLM	7 gennaio 2016
ALTER DEFAULT PRIVILEGES	Puoi utilizzare il comando ALTER DEFAULT PRIVILEGES per definire il set predefinito di privilegi di accesso da applicare agli oggetti creati in futuro dall'utente specificato.	10 dicembre 2015
Compressione di file bzip2	Il comando COPY supporta il caricamento di dati da file che sono stati compressi utilizzando bzip2.	10 dicembre 2015
NULLS FIRST e NULLS LAST	Puoi specificare se una clausola ORDER BY deve classificare NULLS all'inizio o alla fine del set di risultati. Per ulteriori informazioni, consultare Clausola ORDER BY e Riepilogo della sintassi della funzione finestra.	19 Novembre 2015
Parola chiave REGION per CREATE LIBRARY	Se il bucket Amazon S3 che contiene i file della libreria di funzioni definite dall'utente non si trova nella stessa regione AWS del cluster Amazon Redshift, è possibile utilizzare l'opzione REGION per specificare la regione in cui si trovano i dati. Per ulteriori informazi oni, consultare CREATE LIBRARY.	19 Novembre 2015

Modifica	Descrizione	Data della modifica
Funzioni scalari UDF	Ora è possibile creare funzioni scalari definite dall'utente personalizzate per implementare funzionalità di elaborazione non SQL fornite dai moduli supportati da Amazon Redshift nella libreria standard Python 2.7 o dalle funzioni definite dall'utente personalizzate basate sul linguaggio di programmazione Python. Per ulteriori informazioni, consultare Creazione di funzioni definite dall'utente.	11 settembre 2015
Proprietà dinamiche nella configurazione WLM	Il parametro di configurazione WLM ora supporta l'applicazione dinamica di alcune proprietà. Le altre proprietà rimangono modifiche statiche e richiedono il riavvio dei cluster associati per l'applicazione delle modifiche di configurazione. Per ulteriori informazioni, consultare Proprietà di configurazione dinamiche e statiche WLM e Implementazione della gestione del carico di lavoro.	3 agosto 2015
Funzione LISTAGG	Funzione LISTAGG e Funzione finestra LISTAGG restituiscono una stringa creata dalla concatenazione di un set di valori di colonna.	30 luglio 2015
Parametro obsoleto	Il parametro di configurazione max_cursor_result_set_size è obsoleto. La dimensione dei set di risultati del cursore è vincolata in funzione del tipo di nodo del cluster. Per ulteriori informazioni, consultare Vincoli del cursore.	24 luglio 2015
Documentazione del comando COPY modificata	Il riferimento del comando COPY è stato ampiament e modificato per rendere il materiale più intuitivo e accessibile.	15 luglio 2015

Modifica	Descrizione	Data della modifica
COPY per formato Avro	Il comando COPY supporta il caricamento di dati in formato Avro da file di dati in Amazon S3, Amazon EMR e da host remoti mediante SSH. Per ulteriori informazioni, consultare AVRO e Esempi di copia da Avro.	8 luglio 2015
STV_START UP_RECOVE RY_STATE	La tabella di sistema STV_STARTUP_RECOVE RY_STATE registra lo stato delle tabelle che sono temporaneamente bloccate durante le operazion i di riavvio del cluster. Amazon Redshift posiziona un blocco temporaneo sulle tabelle durante la loro elaborazione per risolvere transazioni obsolete dopo un riavvio del cluster.	25 maggio 2015
ORDER BY facoltativa per classificare le funzioni	La clausola ORDER BY è ora facoltativa per determina te funzioni di rango di finestre. Per ulteriori informazi oni, consultare Funzione finestra CUME_DIST, Funzione finestra DENSE_RANK, Funzione finestra RANK, Funzione finestra NTILE, Funzione finestra PERCENT_RANK e Funzione finestra ROW_NUMBE R.	25 maggio 2015
Chiavi di ordinamento interlacciato	Le chiavi di ordinamento interlacciato forniscono lo stesso peso a ogni colonna nella chiave di ordinamen to. L'utilizzo di chiavi di ordinamento interlacciato al posto delle chiavi composte migliora in modo significativo le prestazioni delle query che utilizzan o predicati restrittivi sulle colonne di ordinamen to secondarie, soprattutto per le tabelle di grandi dimensioni. L'ordinamento interlacciato migliora anche le prestazioni globali quando più query filtrano different i colonne nella stessa tabella. Per ulteriori informazi oni, consultare Utilizzo delle chiavi di ordinamento e CREATE TABLE.	11 maggio 2015

Modifica	Descrizione	Data della modifica
Argomento Ottimizzazione delle prestazioni delle query modificato	Ottimizzazione delle prestazioni delle query è stato ampliato per includere nuove query per l'analisi delle prestazioni delle query e altri esempi. È stato inoltre modificato per risultare più chiaro e completo. Best practice di Amazon Redshift per la progettazione di query include più informazioni su come scrivere query per migliorare le prestazioni.	23 marzo 2015
SVL_QUERY_QUEUE_INFO	La vista SVL_QUERY_QUEUE_INFO riepiloga i dettagli delle query che sono state in una coda di query WLM o in una coda di commit.	19 febbraio 2015
SVV_TABLE_INFO	Puoi utilizzare la vista SVV_TABLE_INFO per diagnosticare e risolvere problemi di progettazione delle tabelle che possono influenzare le prestazioni della query, inclusi i problemi relativi a codifica di compressione, chiavi di distribuzione, stile di ordinamento, asimmetria nella distribuzione dei dati, dimensioni delle tabelle e statistiche.	19 febbraio 2015
UNLOAD utilizza la crittografia di file lato server	Il comando UNLOAD ora utilizza automaticamente la crittografia lato server (SSE) di Amazon S3 per crittografare tutti i file di dati di scaricamento. La crittografia lato server aggiunge un altro livello di sicurezza dei dati con un impatto ridotto o inesistente sulle prestazioni.	31 Ottobre 2014
Funzione finestra CUME_DIST	La funzione Funzione finestra CUME_DIST calcola la distribuzione cumulativa di un valore in una finestra o partizione.	31 Ottobre 2014
Funzione MONTHS_BETWEEN	La funzione Funzione MONTHS_BETWEEN determina il numero di mesi tra due date.	31 Ottobre 2014

Modifica	Descrizione	Data della modifica
Funzione NEXT_DAY	La funzione Funzione NEXT_DAY restituisce la data della prima istanza di un determinato giorno che è successiva alla data fornita.	31 Ottobre 2014
Funzione finestra PERCENT_RANK	La funzione Funzione finestra PERCENT_RANK calcola il rango percentuale di una determinata riga.	31 Ottobre 2014
Funzione finestra RATIO_TO_REPORT	La funzione Funzione finestra RATIO_TO_REPORT calcola il rapporto di un valore rispetto alla somma dei valori in una finestra o partizione.	31 Ottobre 2014
Funzione TRANSLATE	La funzione Funzione TRANSLATE sostituisce tutte le occorrenze dei caratteri specificati in una determinata espressione con sostituti specificati.	31 Ottobre 2014
Funzione NVL2	La funzione Funzione NVL2 restituisce uno dei due valori a seconda che l'espressione specificata viene valutata NULL o NOT NULL.	16 ottobre 2014
Funzione finestra MEDIAN	La funzione Funzione finestra MEDIAN calcola il valore mediano per l'intervallo di valori in una finestra o partizione.	16 ottobre 2014
Clausola ON ALL TABLES IN SCHEMA schema_name per i comandi GRANT e REVOKE	I comandi GRANT e REVOKE sono stati aggiornati con una clausola ON ALL TABLES IN SCHEMA schema_name. Questa clausola ti consente di utilizzare un unico comando per modificare i privilegi per tutte le tabelle in uno schema.	16 ottobre 2014

Modifica	Descrizione	Data della modifica
Clausola IF EXISTS per i comandi DROP SCHEMA, DROP TABLE, DROP USER e DROP VIEW	I comandi DROP SCHEMA, DROP TABLE, DROP USER e DROP VIEW sono stati aggiornati con una clausola IF EXISTS. Questa clausola impedisce al comando di apportare modifiche e restituisce un messaggio anziché terminare con un errore se l'oggetto specificato non esiste.	16 ottobre 2014
Clausola IF NOT EXISTS per i comandi CREATE SCHEMA e CREATE TABLE	I comandi CREATE SCHEMA E CREATE TABLE sono stati aggiornati con una clausola IF NOT EXISTS. Questa clausola impedisce al comando di apportare modifiche e restituisce un messaggio anziché terminare con un errore se l'oggetto specifica to esiste.	16 ottobre 2014
Supporto COPY per la codifica UTF-16	Il comando COPY ora supporta il caricamento da file di dati che utilizzano la codifica UTF-16 e la codifica UTF-8. Per ulteriori informazioni, consultare ENCODING.	29 settembre 2014
Nuovo tutorial Gestione dei carichi di lavoro	Tutorial: Configurazione delle code di gestione manuale del carico di lavoro (WLM) descrive il processo di configurazione delle code WLM per migliorare l'elaborazione delle query e l'allocazione delle risorse.	25 settembre 2014
Crittografia AES-128	Il comando COPY ora supporta la crittografia AES-128 nonché la crittografia AES-256 per il caricamento da file di dati crittografati mediante la crittografia lato client di Amazon S3. Per ulteriori informazioni, consultare Caricamento di file di dati crittografati da Amazon S3.	29 settembre 2014

Modifica	Descrizione	Data della modifica
Funzione PG_LAST_UNLOAD_COUNT	La funzione PG_LAST_UNLOAD_COUNT restituisce il numero di righe che sono state elaborate nell'operazione UNLOAD più recente. Per ulteriori informazioni, consultare PG_LAST_UNLOAD_COUNT.	15 settembre 2014
Nuova sezione Risoluzione dei problemi delle query	Risoluzione dei problemi delle query fornisce un riferimento rapido per l'identificazione e la risoluzione di alcuni dei problemi più comuni e più gravi che potrebbero verificarsi durante l'utilizzo di query di Amazon Redshift.	7 luglio 2014
Nuovo tutorial sul caricamento dei dati	Questo Tutorial: Caricamento dei dati da Amazon S3 guida attraverso l'intero processo di caricamento di dati nelle tabelle di database Amazon Redshift a partire dai file di dati in un bucket Amazon S3.	1 luglio 2014
Funzione finestra PERCENTILE_CONT	Funzione finestra PERCENTILE_CONT è una funzione di distribuzione inversa che presuppone un modello di distribuzione continua. Prende un valore percentile e una specifica di ordinamento e restituisce un valore interpolato che rientrerebbe nel valore percentile dato rispetto alla specifica di ordinamento.	30 giugno 2014
Funzione finestra PERCENTILE_DISC	Funzione finestra PERCENTILE_DISC è una funzione di distribuzione inversa che presuppone un modello di distribuzione discreta. Prende un valore percentile e una specifica di ordinamento e restituisce un elemento dall'insieme.	30 giugno 2014
Funzioni GREATEST e LEAST	Le funzioni Funzioni GREATEST e LEAST restituiscono il valore più grande o più piccolo da un elenco di espressioni.	30 giugno 2014

Modifica	Descrizione	Data della modifica
Comando COPY tra regioni	Il comando COPY supporta il caricamento di dati da un bucket Amazon S3 o una tabella Amazon DynamoDB che si trova in una regione diversa dal cluster Amazon Redshift. Per ulteriori informazioni, consultare REGION nel riferimento del comando COPY.	30 giugno 2014
Best practice ampliate	L'argomento Best practice di Amazon Redshift è stato ampliato, riorganizzato e spostato all'inizio della gerarchia di navigazione per renderlo più visibile.	28 maggio 2014
Comando UNLOAD per un singolo file	Il comando UNLOAD può eventualmente scaricare i dati delle tabelle in modo sequenziale in un unico file su Amazon S3 aggiungendo l'opzione PARALLEL OFF. Se la dimensione dei dati supera il limite massimo di 6,2 GB, UNLOAD crea file aggiuntivi.	6 maggio 2014
Funzioni REGEXP	Le funzioni REGEXP_COUNT, REGEXP_INSTR e REGEXP_REPLACE manipolano le stringhe in base alla corrispondenza con il modello di espressione regolare.	6 maggio 2014
COPY da Amazon EMR	Il comando COPY supporta il caricamento di dati direttamente da cluster Amazon EMR. Per ulteriori informazioni, consultare Caricamento di dati da Amazon EMR.	18 aprile 2014
Aumento del limite di simultaneità WLM	Ora puoi configurare WLM per eseguire fino a 50 query simultaneamente nelle code di query definite dall'utente. Questo aumento fornisce agli utenti una maggiore flessibilità per la gestione delle prestazioni del sistema modificando le configurazioni WLM. Per ulteriori informazioni, consultare Implementazione di WLM manuale	18 aprile 2014

Modifica	Descrizione	Data della modifica
Nuovo parametro di configurazione per gestire la dimensione del cursore	Il parametro di configurazione `max_curso r_result_set_size` definisce la dimensione massima, in megabyte, dei dati che possono essere restituiti per set di risultati del cursore di una query più grande. Questo valore di parametro modifica anche il numero di cursori simultanei per il cluster, consenten doti di configurare un valore che aumenta o diminuisce il numero di cursori per il cluster. Per ulteriori informazioni, consulta DECLARE in questa guida e Configurazione della dimensione massima di un set di risultati del cursore nella Guida alla gestione di Amazon Redshift.	28 marzo 2014
COPY dal formato JSON	Il comando COPY supporta il caricamento di dati in formato JSON da file di dati in Amazon S3 e da host remoti mediante SSH. Per ulteriori informazioni, consultare le note per l'utilizzo di COPY dal formato JSON.	25 marzo 2014
Nuova tabella di sistema STL_PLAN_INFO	La tabella STL_PLAN_INFO completa il comando EXPLAIN come un altro modo di esaminare i piani di query.	25 marzo 2014
Nuova funzione REGEXP_SU BSTR	La funzione Funzione REGEXP_SUBSTR restituisce i caratteri estratti da una stringa cercando un modello di espressione regolare.	25 marzo 2014
Nuove colonne per STL_COMMI T_STATS	La tabella STL_COMMIT_STATS include due nuove colonne, ovvero `numxids` e `oldestxid` .	6 marzo 2014
Supporto di COPY per gzip e lzop mediante SSH	Il comando COPY supporta la compressione gzip e lzop per il caricamento di dati mediante una connessio ne SSH.	13 febbraio 2014

Modifica	Descrizione	Data della modifica
Nuove funzioni	La funzione Funzione finestra ROW_NUMBER restituisce il numero della riga corrente. La funzione Funzione STRTOL converte un'espressione di stringa di un numero della base specificata nel valore intero equivalente. PG_CANCEL_BACKEND e PG_TERMINATE_BACKEND consentono agli utenti di annullare query e connessioni delle sessioni. La funzione LAST_DAY è stata aggiunta per la compatibilità con Oracle.	13 febbraio 2014
Nuova tabella di sistema	La tabella di sistema STL_COMMIT_STATS fornisce parametri correlati alle prestazioni di commit, tra cui la tempistica delle varie fasi di commit e il numero di blocchi sottoposti a commit.	13 febbraio 2014
FETCH con cluster a nodo singolo	Quando utilizzi un cursore su un cluster a nodo singolo, il numero massimo di righe che possono essere recuperate mediante il comando FETCH è 1000. FETCH FORWARD ALL non è supportato per i cluster a nodo singolo.	13 febbraio 2014
Strategia di ridistribuzione DS_DIST_ALL_INNER	DS_DIST_ALL_INNER nell'output del piano Explain indica che tutta la tabella interna è stata ridistribuita a una singola sezione in quanto la tabella esterna utilizza DISTSTYLE ALL, Per ulteriori informazioni, consultare Esempi di tipo di join e Valutazione del piano di query.	13 gennaio 2014
Nuove tabelle di sistema per query	Amazon Redshift ha aggiunto nuove tabelle di sistema che i clienti possono utilizzare per valutare l'esecuzione delle query per l'ottimizzazione e la risoluzione dei problemi. Per ulteriori informazioni, consultare SVL_COMPILE, STL_SCAN, STL_RETURN, STL_SAVE STL_ALERT_EVENT_LOG.	13 gennaio 2014

Modifica	Descrizione	Data della modifica
Cursori a nodo singolo	I cursori sono ora supportati per i cluster a nodo singolo. Un cluster a nodo singolo può avere due cursori aperti contemporaneamente, con un set di risultati massimo di 32 GB. Su un cluster a nodo singolo, consigliamo di impostare il parametro della dimensione della cache ODBC su 1.000. Per ulteriori informazioni, consultare DECLARE.	13 dicembre 2013
Stile di distribuzione ALL	La distribuzione ALL può ridurre considerevolmente i tempi di esecuzione per certi tipi di query. Quando una tabella utilizza lo stile di distribuzione ALL, una copia della tabella viene distribuita a ogni nodo. Poiché la tabella è collocata effettivamente con ogni altra tabella, non è necessaria alcuna ridistribuzione durante l'esecuzione della query. La distribuzione ALL non è appropriata per tutte le tabelle in quanto aumenta i bisogni di storage e il tempo di caricamento. Per ulteriori informazioni, consultare Utilizzo degli stili di distribuzione dati.	11 Novembre 2013
COPY da host remoti	Oltre a caricare tabelle da file di dati su Amazon S3 e da tabelle Amazon DynamoDB, il comando COPY può caricare dati di testo da cluster Amazon EMR, istanze Amazon EC2 e altri host remoti tramite le connessioni SSH. Amazon Redshift utilizza più connessioni SSH simultanee per leggere e caricare i dati in parallelo. Per ulteriori informazioni, consultare Caricamento di dati da host remoti.	11 Novembre 2013
Percentuale di memoria WLM utilizzata	Puoi bilanciare il carico di lavoro indicando una specifica percentuale di memoria per ogni coda nella configurazione WLM. Per ulteriori informazioni, consultare Implementazione di WLM manuale.	11 Novembre 2013

Modifica	Descrizione	Data della modifica
APPROXIMATE COUNT(DISTINCT)	L'esecuzione delle query che utilizzano APPROXIMATE COUNT(DISTINCT) è molto più rapida, con un errore relativo di circa il 2%. La funzione APPROSSIMATE COUNT (DISTINCT) utilizza un HyperLogLog algoritmo. Per ulteriori informazioni, consultare la Funzione COUNT.	11 Novembre 2013
Nuove funzioni SQL per recuperare dettagli di query recenti	Quattro nuove funzioni SQL recuperano dettagli sulle query recenti e i comandi COPY. Le nuove funzioni facilitano l'esecuzione di query su tabelle di log di sistema e in molti casi forniscono dettagli necessari senza dover accedere alle tabelle di sistema. Per ulteriori informazioni, consultare PG_BACKEND_PID, PG_LAST_COPY_ID, PG_LAST_COPY_COUNT, PG_LAST_QUERY_ID.	1 Novembre 2013
Opzione MANIFEST per UNLOAD	L'opzione MANIFEST per il comando UNLOAD completa l'opzione MANIFEST per il comando COPY. L'utilizzo dell'opzione MANIFEST con UNLOAD crea automaticamente un file manifest che elenca esplicitamente i file di dati creati in Amazon S3 dall'operazione di scaricamento. Puoi quindi utilizzare lo stesso file manifest con un comando COPY per caricare i dati. Per ulteriori informazioni, consultare Scaricamento dei dati in Amazon S3 e Esempi di UNLOAD.	1 Novembre 2013
Opzione MANIFEST per COPY	È possibile utilizzare l'opzione MANIFEST con il comando COPY per elencare esplicitamente i file di dati che verranno caricati da Amazon S3.	18 ottobre 2013

Modifica	Descrizione	Data della modifica
Tabelle di sistema per la risoluzione dei problemi delle query	Aggiunta della documentazione relativa alle tabelle di sistema utilizzate per la risoluzione dei problemi delle query. La sezione Viste STL per la registrazione ora contiene la documentazione per le seguenti tabelle di sistema: STL_AGGR, STL_BCAST, STL_DIST, STL_DELETE, STL_HASH, STL_HASHJOIN, STL_INSERT, STL_LIMIT, STL_MERGE, STL_MERGEJOIN, STL_NESTLOOP, STL_PARSE, STL_PROJECT, STL_SCAN, STL_SORT, STL_UNIQUE, STL_WINDOW.	3 ottobre 2013
Funzione CONVERT_TIMEZONE	La funzione Funzione CONVERT_TIMEZONE converte un timestamp di un fuso orario in un altro, con la possibilità di passare automaticamente all'ora legale.	3 ottobre 2013
Funzione SPLIT_PART	La funzione Funzione SPLIT_PART divide una stringa sul delimitatore specificato e restituisce la parte nella posizione specificata.	3 ottobre 2013
Tabella di sistema STL_USERLOG	La funzione STL_USERLOG registra i dettagli per le modifiche che si hanno quando un utente di database viene creato, modificato o eliminato.	3 ottobre 2013
Codifica di colonna LZO e compressione di file LZOP	La codifica di compressione di colonna LZO combina un rapporto di compressione molto elevato e buone prestazioni. L'esecuzione del comando COPY da Amazon S3 supporta il caricamento da file che sono stati compressi utilizzando la compressione LZOP.	19 settembre 2013
JSON, espressioni regolari e cursori	Aggiunta del supporto per l'analisi di stringhe JSON, la corrispondenza con modelli mediante espressioni regolari e l'utilizzo di cursori per recuperare set di dati di grandi dimensioni via una connessione ODBC. Per ulteriori informazioni, consultare Funzioni JSON, Condizioni di corrispondenza di modelli e DECLARE.	10 settembre 2013

Modifica	Descrizione	Data della modifica
Opzione ACCEPTINV CHAR per COPY	Puoi caricare senza problemi dati che contengon o caratteri UTF-8 non validi specificando l'opzione ACCEPTINVCHAR con il comando COPY.	29 agosto 2013
Opzione CSV per COPY	Il comando COPY ora supporta il caricamento da file di input in formato CSV.	9 agosto 2013
CRC32	La funzione Funzione CRC32 esegue controlli di ridondanza ciclici.	9 agosto 2013
Caratteri jolly WLM	WLM supporta i caratteri jolly per l'aggiunta di gruppi di utenti e di query alle code. Per ulteriori informazioni, consultare Caratteri jolly.	1 agosto 2013
Timeout WLM	Per limitare il tempo di utilizzo delle query in una determinata coda WLM, puoi impostare il valore di timeout WLM per ogni coda. Per ulteriori informazioni, consultare Timeout WLM.	1 agosto 2013
Nuove opzioni "auto" e "epochsecs" di COPY	Il comando COPY esegue il riconoscimento automatic o dei formati di data e ora. I nuovi formati di ora, "epochsecs" e "epochmillisecs" consentono al comando COPY di caricare dati nel formato epoch.	25 luglio 2013
Funzione CONVERT_T IMEZONE	La funzione Funzione CONVERT_TIMEZONE converte un timestamp da un fuso orario a un altro.	25 luglio 2013
Funzione FUNC_SHA1	La funzione Funzione FUNC_SHA1 converte una stringa mediante l'algoritmo SHA1.	15 luglio 2013
max_execu tion_time	Per limitare il tempo di esecuzione delle query, puoi impostare il parametro max_execution_time nella configurazione WLM. Per ulteriori informazioni, consultare Modifica della configurazione WLM.	22 luglio 2013

Modifica	Descrizione	Data della modifica
Caratteri UTF-8 a quattro byte	Il tipo di dati VARCHAR ora supporta caratteri UTF-8 a quattro byte. I caratteri UTF-8 a cinque byte o più non sono supportati. Per ulteriori informazioni, consultare Storage e intervalli.	18 luglio 2013
SVL_QERROR	La vista di sistema SVL_QERROR è obsoleta.	12 luglio 2013
Cronologia dei documenti modificata	La pagina Document History (Cronologia dei documenti) ora include la data di aggiornamento della documentazione.	12 luglio 2013
STL_UNLOAD_LOG	STL_UNLOAD_LOG registra i dettagli di un'operazione di scaricamento.	5 luglio 2013
Parametro delle dimensioni di recupero JDBC	Per evitare errori di memoria insufficiente sul lato client durante il recupero di grandi set di dati mediante JDBC, è possibile consentire al client di recuperare i dati in batch impostando il parametro relativo alle dimensioni di recupero JDBC. Per ulteriori informazioni, consultare Impostazione del parametro delle dimensioni del recupero JDBC.	27 giugno 2013
File crittografati UNLOAD	UNLOAD ora supporta lo scaricamento di dati di tabelle in file crittografati su Amazon S3.	22 maggio 2013
Credenziali temporanee	COPY e UNLOAD ora supportano l'utilizzo di credenziali temporanee.	11 Aprile 2013
Aggiunta di precisioni	Chiarite e ampliate discussioni sulla progettazione di tabelle e sul caricamento di dati.	14 febbraio 2013
Aggiunte best practice	Aggiunto Best practice di Amazon Redshift per la progettazione di tabelle e Best practice di Amazon Redshift per il caricamento di dati.	14 febbraio 2013

Modifica	Descrizione	Data della modifica
Precisazioni sui vincoli relativi alle password	Precisazioni sui vincoli relativi alle password per CREATE USER e ALTER USER, varie revisioni minori.	14 febbraio 2013
Nuova guida	Questa è la prima versione della Guida per gli sviluppatori di Amazon Redshift.	14 febbraio 2013

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.