



Amazon Redshift でマテリアライズドビューを使用する

AWS 規範ガイド



AWS 規範ガイド: Amazon Redshift でマテリアライズドビューを使用する

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

序章	1
概要	1
ターゲットを絞ったビジネス成果	1
マテリアライズドビューを理解する	2
ビュー	2
マテリアライズドビュー	2
ビュータイプの比較	3
マテリアライズドビューの作成	5
マテリアライズドビューのクエリ	7
マテリアライズドビューの更新	8
自動クエリ書き換え	10
クエリ書き換えの仕組み	10
マテリアライズドビューの利点	12
リソース	13
ドキュメント履歴	14
.....	xv

Amazon Redshift でマテリアライズドビューを使用する

Ethan Stark、Kelly Ragan、Srinivasan Krishnasamy (Amazon Web Services (AWS))

2022 年 12 月 ([ドキュメント履歴](#))

概要

データウェアハウスのアプリケーションでは多くの場合、大規模なテーブルで複雑なクエリを実行する必要があります。これらのクエリの処理に時間と費用がかかることもあります。例えば、Amazon Redshift でクエリを実行すると、リーダーノードはクエリの解析、検証、計画、最適化、実行を行います。このプロセスでは、ウォールクロック時間、CPU 時間、メモリを大量に消費することで、機会費用と実際のコストが増加する可能性があります。

このガイドでは、Amazon Redshift のマテリアライズドビューを使用してクエリ、特に予測可能で頻繁に繰り返されるクエリを高速化する方法について説明します。マテリアライズドビューは事前に計算された結果セットを保存することでクエリ時間を短縮するため、基となるベーステーブルに直接アクセスする必要がなくなります。クエリ時間を短縮することで、クエリ処理のコストを削減できるため、アプリケーションをコスト効率よくスケールできます。このガイドはデータエンジニア、データアーキテクト、データアナリストを対象としています。

ターゲットを絞ったビジネス成果

このガイドを使用すると、次のようなビジネス成果を達成できます。

- Amazon Redshift クエリの処理コストを削減
- アプリケーションを効率的かつ費用対効果の高い方法で拡張
- より価値の高いタスクに時間を割けるようユーザーを解放

マテリアライズドビューを理解する

ビュー

ビューは SQL SELECT クエリの結果セットに基づく仮想テーブルです。仮想テーブルにはクエリ式から取得したデータが含まれていますが、結果はディスクには保存されません。ビューを使用すると、クエリを実行するたびに元のテーブルからデータが取得されるため、常に最新のデータを取得できます。1 つ以上のベーステーブルまたはビューから 1 つのビューを作成できます。元のベーステーブルをクエリするのと同じ方法でビューをクエリすることもできます。

次の例では、クエリでビューを作成する方法を示しています。

```
CREATE VIEW tickets_view
AS
  select e.eventname,
         sum(s.price) as total_sales
  from sales s
  join event e
    on e.eventid = s.eventid
  group by e.eventname;
```

次の例では、ビューをクエリする方法を示しています。

```
select eventname,
       total_sales
from ticket_view
where eventname = 'Gotterburg';
```

マテリアライズドビュー

マテリアライズドビューは、クエリの結果を含むデータベースオブジェクトです。例えば、マテリアライズドビューは次のいずれかになります。

- リモートにあるデータのローカルコピー
- テーブルまたは結合結果の行または列のサブセット
- 集計関数を使用したサマリー

ビュータイプの比較

次の表は、ビューとマテリアライズドビューの違いをまとめたものです。

キー	表示	マテリアライズドビュー
定義	データを保存せず、代わりに定義済みの SQL クエリを実行してデータベース内の 1 つ以上のテーブルからデータを取得する仮想テーブル	編集可能な SQL クエリによって定義されるが、クエリの結果はディスクに格納される仮想テーブル
ストレージ	クエリ式の結果はディスクには保存されず、クエリ式のみがディスクに保存されます	クエリ式とクエリ式の結果は両方ともディスクに保存されます
実行	ビューを定義するクエリは、1 つのクエリでビューが参照されるたびに実行されます	クエリの結果はディスクに保存されます。ユーザーがマテリアライズドビューからデータを取得しようとするたびにクエリ式が実行されるわけではありません
データの最新性	ベーステーブルから常に最新の更新値が提供されます	データベース内で値が変更されても、最新の更新値は提供されません
コスト	ストレージコストなし	ストレージコストあり
設計	標準ビューを作成するには、次のことを行う必要があります。 ・ 基になるテーブルにアクセスする ・ 標準の SELECT ステートメントを使用する	マテリアライズドビューを作成するには、次のことを行う必要があります。 ・ 基になるテーブルにアクセスする ・ 標準の SELECT ステートメントを使用する

オプションで、以下のように指定できます。

Amazon Simple Storage Service (Amazon S3) に保存されている自動または手動クラスタースナップショットに、マテリアライズドビューを含むかどうか

マテリアライズドビューのデータを分散およびソートする方法

マテリアライズドビューを、ベーステーブルからの最新の変更で自動的に更新する必要があるかどうか

使用方法

データへのアクセスや更新の頻度が低い場合

データへのアクセスや更新が頻繁な場合

マテリアライズドビューの作成

以下のうちの 1 つ以上に基づいて、マテリアライズドビューを作成できます。

- Amazon Redshift テーブル
- [Amazon Redshift Spectrum](#) または [フェデレーションクエリ](#) を使用して作成できる外部テーブル。
- その他のマテリアライズドビュー

マテリアライズドビューを作成する際、Amazon Redshift は、ユーザーが指定した SQL ステートメントを実行し、ベーステーブルまたはテーブルからデータを収集します。次に、Amazon Redshift は結果セットを保存します。

以下の図は、mv_total_orders と呼ばれるマテリアライズドビューの概要を示しています。このビューは、顧客と注文という 2 つのベーステーブルを使用する SQL クエリによって定義されます。

Customer

cust_id	first_name
1	Akua
2	Ana
3	Jane
4	Mary
5	Paulo
6	Richard
7	Saanvi
8	Wei

Order

order_id	customer_id	amount
1	5	100
2	3	200
3	2	500
4	1	300
5	2	200
6	5	300
7	6	200



```
CREATE MATERIALIZED VIEW mv_total_orders
AS
  SELECT c.cust_id,
         c.first_name,
         sum(o.amount) as total_amount
  FROM orders o
  JOIN customer c
    ON c.cust_id = o.customer_id
  GROUP BY c.cust_id,
           c.first_name;
```

mv_total_orders

cust_id	first_name	total_amount
1	Akua	300
2	Ana	700
3	Jane	200
5	Paulo	400
6	Richard	200

マテリアライズドビューのクエリ

マテリアライズドビューのクエリを実行すると、マテリアライズドビューの計算済みのデータに直接アクセスします。テーブルやスタンダードビューのデータソースとしてマテリアライズドビューの名前を参照することにより、任意の SQL クエリでマテリアライズドビューを使用できます。

例えば、このガイドの「[Creating materialized views](#)」にある mv_total_orders のマテリアライズドビューの例を示した図について考えてみましょう。mv_total_orders のクエリを作成したい場合 (これにより、注文総額が 500 USD を超える顧客のリストが返されます)、次の標準クエリを実行できます。

```
statement. SELECT c.cust_id,
                  c.first_name,
                  sum(o.amount) as total_amount
FROM orders o
JOIN customer c
  ON c.cust_id = o.customer_id
GROUP BY c.cust_id,
         c.first_name
HAVING sum(o.amount) > 500;
```

ただし、前述のクエリは速度が最適化されていません。代わりに次のクエリの実行を推奨します。

```
SELECT cust_id,
       first_name,
       total_amount
FROM mv_total_orders
WHERE total_amount > 500;
```

推奨するクエリは、クエリの結果が事前計算されているため実行時間が大幅に短縮され、基となるテーブル (顧客および注文) にアクセスする必要もありません。Amazon Redshift は、mv_total_orders から直接結果を返すことができます。

[重要]

クエリがマテリアライズドビューにアクセスする際は、直近の更新時点でマテリアライズドビューに保存されているデータのみを参照します。そのためクエリは、マテリアライズドビューに対応するベーステーブルからの最新の変更すべてを参照しない場合があります。

マテリアライズドビューの更新

マテリアライズドビューにはクエリ結果のスナップショットが含まれます。Amazon Redshift を定期的に更新するように設定しない限り、マテリアライズドビューは定期的に更新されません。REFRESH MATERIALIZED VIEW ステートメントを使用して、マテリアライズドビューのデータを随時手動で更新できます。このコマンドはベーステーブルで行われた変更を特定し、その変更をマテリアライズドビューに適用します。

マテリアライズドビューを更新する方法は、手動更新と自動更新 ([自動リフレッシュ](#)) の 2 種類があります。次のクエリ例は、マテリアライズドビューを手動で更新する方法を示しています。

```
REFRESH MATERIALIZED VIEW mv_total_orders;
```

マテリアライズドビューを自動更新するには、以下の例にあるように、AUTO REFRESH YES 句を CREATE MATERIALIZED VIEW のステートメントに追加します。

```
CREATE MATERIALIZED VIEW mv_total_orders
AUTO REFRESH YES -- Add this clause to auto refresh the MV
AS
SELECT c.cust_id,
       c.first_name,
       sum(o.amount) as total_amount
FROM orders o
JOIN customer c
  ON c.cust_id = o.customer_id
GROUP BY c.cust_id,
         c.first_name;
```

Amazon Redshift は、ベーステーブルが変更されると、可及的速やかにマテリアライズドビューを自動更新します。更新を処理する際、クラスターにあるアクティブなワークロードの影響を最小限に抑えるため、Amazon Redshift は次の要素を考慮します。

- 現在のシステム負荷
- 更新に必要なリソース
- 利用可能なクラスターリソース
- マテリアライズドビューの使用頻度

Amazon Redshift は自動更新よりもワークロードを優先し、ユーザーのワークロードのパフォーマンスを維持するために自動更新を停止する場合があります。この方法では一部のマテリアライズドビューの更新が遅れる可能性があることに注意してください。更新ステータスについては、[SVL_MV_REFRESH_STATUS](#) ビューで確認できます。このビューには、ユーザーが開始したクエリまたは自動更新されたクエリが記録されます。

自動クエリ書き換え

Amazon Redshift では、パフォーマンス向上のために使用される既存のマテリアライズドビューを明示的に参照しない SQL クエリを、自動的に書き換えることができます。マテリアライズドビューを直接クエリするとクエリの処理時間は短縮されますが、クエリステートメントを変更する必要があります。マテリアライズドビューを使用してクエリを高速化する場合、マテリアライズドビューを体系的かつ自動的に使用してクエリに応答する方法を検討することが重要です。透過的な書き換えを使用すると、既存の SQL ステートメントに影響を与えることなく、インデックスと同じ方法でマテリアライズドビューを追加または削除できます。Amazon Redshift は、クエリを自動で書き換える際に、最新のマテリアライズドビューのみを使用します。

クエリ書き換えの仕組み

次の SQL クエリについて考えてみましょう。このクエリは、顧客 (1,000 万行以上) と注文 (100 億行以上) の 2 つのテーブルを結合します。

```
SELECT c.cust_id,
       c.first_name,
       sum(o.amount) as total_amount
FROM orders o
JOIN customer c
  ON c.cust_id = o.customer_id
GROUP BY c.cust_id,
         c.first_name
HAVING sum(o.amount) > 500;
```

このクエリは 2 つの大きなテーブルを結合し、o.amount 列にその SUM 集約を適用します。それから結果をフィルタリングして、500 USD 以上の注文を行った顧客のみを表示します。このクエリは大量のリソースを消費する可能性があります。

例えば、このガイドの「[マテリアライズドビューの作成](#)」セクションから、mv_total_orders のマテリアライズドビューについて考えてみましょう。このビューが作成されると、Amazon Redshift は前のクエリを次のように自動的に書き換えます。

```
SELECT cust_id,
       first_name,
       total_amount
FROM mv_total_orders
```

```
WHERE total_amount > 500;
```

この新しいクエリは、2 つの大きなテーブルを結合する代わりにマテリアライズドビューを使用するため、はるかに高速に実行されます。

クエリの書き換えによってマテリアライズドビューが広く採用されるようになったのは、以下の理由からです。

1. マテリアライズドビューはクエリの透過的な書き換えをサポートします。クエリの書き換えは、テーブルで表される SQL ステートメントを 1 つ以上のマテリアライズドビューにアクセスするステートメントに変換します。この変換はエンドユーザーには分かりやすく、介入する必要も、SQL ステートメントのマテリアライズドビューを参照する必要もありません。
2. マテリアライズドビューは、キャッシュされた共通結果セットの適用を簡素化し、事前計算などクросクエリの最適化をサポートします。
3. データウェアハウスやデータ統合シナリオでは、マテリアライズドビューは外部テーブルの結果をマテリアライズして、ローカルレプリカや読み取り/書き込み分割を実装することで、複数のデータソース間の違いを隠すことができます。
4. クエリの書き換え機能とマテリアライズドビューの自動作成を組み合わせることで、データベースの自律性が向上します。

マテリアライズドビューの利点

マテリアライズドビューを使用する利点はいくつかあります。

- **更新が少ない** — 標準ビューは物理的にマテリアライズドされません。つまり、標準ビューを定義するクエリは、ビューがクエリで参照されるたびに実行されます。対照的に、マテリアライズドビューは通常のビューと同様にクエリ式の結果として事前に計算され、ディスクに (オブジェクトと同様に) 格納されます。標準ビューとは異なり、マテリアライズドビューは使用されるたびに更新されません。
- **応答時間の短縮** — マテリアライズドビューは、ビューに比べて応答が速くなります。これは、マテリアライズドビューは事前に計算されているため、マテリアライズドビューを作成するクエリやクエリの結合の解決で時間を無駄にしないためです。
- **ストアド SQL ステートメント** — マテリアライズドビューの代わりにロールアップテーブルまたは集計テーブルを使用できます。ロールアップテーブルは事前に計算されてディスクに保存されますが (マテリアライズドビューと同様) データベースに SQL ステートメントは保存されません。マテリアライズドビューには SQL ステートメントが保存されます。
- **更新が簡単** — マテリアライズドビューは簡単に更新できます。REFRESH MATERIALIZED VIEW コマンドを実行するだけです。
- **自動クエリ書き換え** — クエリオプティマイザーは、SQL ステートメントでマテリアライズドビューが明示的に使用されていない場合でも、既存のマテリアライズドビューからデータを取得するように SQL ステートメントを書き換えることができます。

リソース

リファレンス

- [Amazon Redshift でのマテリアライズドビューの作成](#)
- [マテリアライズドビューのクエリの実行](#)
- [マテリアライズドビューを使用するための自動クエリ書き換え](#)
- [マテリアライズドビューの更新](#)

ツール

- [Amazon Redshift クエリエディタ v2.0](#)

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、[RSS フィード](#) をサブスクライブできます。

変更	説明	日付
初版発行	—	2022 年 12 月 15 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。