



Amazon Redshift에서 구체화된 뷰 사용

AWS 권장 가이드



AWS 권장 가이드: Amazon Redshift에서 구체화된 뷰 사용

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

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

소개	1
개요	1
목표 비즈니스 성과	1
구체화된 뷰 이해	2
보기	2
구체화된 뷰	2
뷰 유형 비교	3
구체화된 뷰 생성	5
구체화된 뷰 쿼리	7
구체화된 뷰 새로 고침	8
자동 쿼리 다시 작성	10
쿼리 다시 작성 작동 방식	10
구체화된 뷰의 장점	12
리소스	13
문서 기록	14
.....	xv

Amazon Redshift에서 구체화된 뷰 사용

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

2022년 12월([문서 기록](#))

개요

데이터 웨어하우스 애플리케이션에서는 대형 테이블에 대해 복잡한 쿼리를 수행해야 하는 경우가 많습니다. 이러한 쿼리를 처리하는 데 많은 시간과 비용이 들 수 있습니다. 예를 들어, Amazon Redshift에서 쿼리를 실행할 때 리더 노드가 쿼리를 구문 분석, 검증, 계획, 최적화 및 실행합니다. 이 프로세스는 월 클럭 시간, CPU 시간 및 메모리를 상당히 많이 소모하여 기회 비용과 실제 비용을 늘릴 수 있습니다.

이 가이드는 Amazon Redshift에서 구체화된 뷰를 사용하여 쿼리, 특히 예측 가능하고 자주 반복되는 쿼리의 속도를 높이는 방법을 보여줍니다. 구체화된 뷰는 미리 계산된 결과 세트를 저장하여 쿼리 시간을 단축하므로 기본 테이블에 직접 액세스할 필요가 없습니다. 이렇게 쿼리 시간이 단축되면 쿼리 처리 비용이 줄어 비용 효율적으로 애플리케이션 규모를 조정할 수 있습니다. 이 가이드는 데이터 엔지니어, 데이터 아키텍트 및 데이터 분석가를 대상으로 합니다.

목표 비즈니스 성과

이 가이드를 사용하여 다음과 같은 비즈니스 성과를 달성할 수 있습니다.

- Amazon Redshift 쿼리 처리 비용 절감
- 효율적이고 경제적으로 애플리케이션 확장
- 사용자가 더 가치 있는 작업에 시간을 할애할 수 있도록 지원

구체화된 뷰 이해

보기

뷰는 SQL SELECT 쿼리의 결과 세트를 기반으로 하는 가상 테이블입니다. 가상 테이블에는 쿼리 식에서 검색된 데이터가 포함되지만 결과는 디스크에 저장되지 않습니다. 뷰를 사용하면 쿼리를 실행할 때마다 쿼리가 원본 테이블에서 데이터를 가져오기 때문에 항상 최신 데이터를 얻을 수 있습니다. 하나 이상의 기본 테이블 또는 뷰에서 뷰를 생성할 수 있습니다. 원본 기본 테이블을 쿼리하는 것과 같은 방식으로 뷰를 쿼리할 수도 있습니다.

다음 예제 쿼리에서는 뷰를 생성하는 방법을 보여줍니다.

```
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;
```

다음 예제 쿼리에서는 뷰를 생성하는 방법을 보여줍니다.

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

구체화된 뷰

구체화된 뷰는 쿼리 결과를 포함하는 데이터베이스 객체입니다. 예를 들어, 구체화된 뷰는 다음 중 하나일 수 있습니다.

- 원격에 위치한 데이터의 로컬 사본
- 테이블 또는 조인 결과의 행 또는 열 하위 세트
- 집계 함수를 사용한 요약

뷰 유형 비교

다음 표에는 뷰와 구체화된 뷰의 차이점이 요약되어 있습니다.

Key(키)	보기	구체화된 뷰
정의	데이터를 저장하지 않고 대신 정의된 SQL 쿼리를 실행하여 데이터베이스의 하나 이상의 테이블에서 데이터를 가져오는 가상 테이블	편집 가능한 SQL 쿼리로 정의되지만 쿼리 결과가 디스크에 저장되는 가상 테이블
스토리지	쿼리 식의 결과는 디스크에 저장되지 않고 쿼리 식만 디스크에 저장됨	쿼리 식과 쿼리 식 결과 모두 디스크에 저장됨
Run	뷰를 정의하는 쿼리는 쿼리에서 해당 뷰가 참조될 때마다 실행됨	쿼리 결과는 디스크에 저장되며, 사용자가 구체화된 뷰에서 데이터를 가져오려고 할 때마다 쿼리 식이 실행되지는 않음
데이터 최신성	항상 기본 테이블에서 업데이트된 최신 값 제공	데이터베이스에서 업데이트된 최신 값이 변경되는 경우 해당 값을 제공하지 않음
비용	스토리지 비용 없음	스토리지 비용 있음
설계	표준 뷰를 만들려면 · 기본 테이블에 액세스할 수 있어야 함 · 표준 SELECT 문을 사용해야 함	구체화된 뷰를 생성하려면 · 기본 테이블에 액세스할 수 있어야 함 · 표준 SELECT 문을 사용해야 함 · 필요한 경우 다음을 지정할 수 있습니다. · 구체화된 뷰가 Amazon Simple Storage Service(S3)에

		저장된 자동 및 수동 클러스터 스냅샷에 포함되는지 여부 · 구체화된 뷰의 데이터가 배포되고 정렬되는 방법 · 구체화된 뷰가 기본 테이블의 최신 변경 사항으로 자동으로 새로 고쳐져야 하는지 여부
사용법	데이터에 자주 액세스하거나 업데이트하지 않을 때	데이터에 자주 액세스하거나 업데이트할 때

구체화된 뷰 생성

다음 중 하나를 기반으로 구체화된 뷰를 생성할 수 있습니다.

- Amazon Redshift 테이블
- [Amazon Redshift Spectrum](#) 또는 [연합 쿼리](#)를 사용하여 생성할 수 있는 외부 테이블
- 다른 구체화된 뷰

구체화된 뷰를 생성할 때 Amazon Redshift는 사용자 지정 SQL 문을 실행하여 하나 이상의 기본 테이블에서 데이터를 수집합니다. 그런 다음 결과 세트를 저장합니다.

다음 그림에서는 mv_total_orders라는 구체화된 뷰를 간략히 보여줍니다. 이 뷰는 customer와 order라는 2개의 기본 테이블을 사용하는 SQL 쿼리로 정의됩니다.

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

구체화된 뷰 쿼리

구체화된 뷰를 쿼리할 때 구체화된 뷰에서 미리 계산된 데이터에 직접 액세스합니다. 구체화된 뷰는 테이블 또는 표준 뷰와 같이 구체화된 뷰 이름을 데이터 소스로 참조하여 모든 SQL 쿼리에서 사용할 수 있습니다.

예를 들어, 이 가이드의 [구체화된 뷰 생성](#) 섹션에 있는 mv_total_orders 구체화된 뷰 예제 그림을 살펴보세요. mv_total_orders에 대한 쿼리(총 주문 금액이 500 USD를 넘는 고객 목록 반환)를 작성하려는 경우 다음 표준 쿼리를 실행할 수 있습니다.

```
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;
```

그러나 위의 쿼리는 속도에 최적화되지 않았습니다. 대신 다음 쿼리를 실행하는 것이 좋습니다.

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

쿼리 결과가 사전 계산되고 기본 테이블(customer 및 order)에 액세스할 필요가 없기 때문에 권장 쿼리가 훨씬 더 빨리 실행됩니다. Amazon Redshift는 mv_total_orders에서 직접 결과를 반환할 수 있습니다.

중요

쿼리가 구체화된 뷰에 액세스하면 쿼리는 가장 최근 새로 고침으로 구체화된 뷰에 저장된 데이터만 볼 수 있습니다. 따라서 쿼리는 구체화된 뷰의 해당 기본 테이블에서 모든 최신 변경 사항을 볼 수 없습니다.

구체화된 뷰 새로 고침

구체화된 뷰에는 쿼리 결과의 스냅샷이 포함됩니다. 주기적으로 업데이트하도록 Amazon Redshift를 구성하지 않는 한 구체화된 뷰는 주기적으로 업데이트되지 않습니다. 구체화된 뷰의 데이터를 수동으로 새로 고치고 업데이트하기 위해 언제든지 REFRESH MATERIALIZED VIEW 문을 사용할 수 있습니다. 이 명령은 기본 테이블에서 발생한 변경 사항을 식별한 다음 해당 변경 사항을 구체화된 뷰에 적용합니다.

구체화된 뷰를 새로 고치는 방법에는 수동 새로 고침과 자동 새로 고침([자동 새로 고침](#)) 두 가지가 있습니다. 다음 예제 쿼리는 구체화된 뷰를 수동으로 새로 고치는 방법을 보여줍니다.

```
REFRESH MATERIALIZED VIEW mv_total_orders;
```

구체화된 뷰를 자동으로 새로 고치려면 다음 예와 같이 CREATE MATERIALIZED VIEW 문에 AUTO REFRESH YES 절을 추가합니다.

```
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 Redshift는 기본 테이블이 변경된 후 가능한 한 빨리 구체화된 뷰를 자동으로 새로 고칩니다. 새로 고침을 처리할 때 클러스터의 활성 워크로드가 미치는 영향을 최소화하기 위해 Amazon Redshift는 다음 요소를 고려합니다.

- 현재 시스템 로드
- 새로 고침에 필요한 리소스
- 사용 가능한 클러스터 리소스
- 구체화된 뷰의 사용 빈도

Amazon Redshift는 자동 새로 고침보다 워크로드에 우선순위를 두며 사용자 워크로드의 성능을 유지하기 위해 자동 새로 고침을 중지할 수 있습니다. 이 접근 방식은 일부 구체화된 뷰의 새로 고침을 지연시킬 수 있다는 점에 유의하세요. 새로 고침 상태는 [SVL_MV_REFRESH_STATUS](#) 뷰에서 확인할 수 있습니다. 이 뷰에는 사용자가 시작한 쿼리나 자동으로 새로 고침 쿼리가 기록됩니다.

자동 쿼리 다시 작성

Amazon Redshift는 성능 향상에 사용되는 기존의 구체화된 뷰를 명시적으로 참조하지 않는 SQL 쿼리를 자동으로 다시 작성할 수 있습니다. 구체화된 뷰를 직접 쿼리하면 쿼리 처리 시간이 단축되지만 쿼리 문을 수정해야 합니다. 구체화된 뷰를 사용하여 쿼리 속도를 높이는 경우 자동으로 구체화된 뷰를 체계적으로 사용하여 쿼리에 응답하는 방법을 고려하는 것이 중요합니다. 투명한 다시 작성을 사용하면 기존 SQL 문에 영향을 주지 않고 인덱스와 동일한 방식으로 구체화된 뷰를 추가하거나 삭제할 수 있습니다. Amazon Redshift는 쿼리를 자동으로 다시 작성할 때 최신의 구체화된 뷰만 사용합니다.

쿼리 다시 작성 작동 방식

다음 SQL 쿼리를 고려해보세요. 쿼리는 customer(행 1,000만 개 이상)과 order(행 100억 개 이상)라는 두 테이블을 조인합니다.

```
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;
```

이 쿼리는 2개의 대형 테이블을 조인하고 o.amount 열에 집계 집계를 적용한 다음 500 USD 이상 주문한 고객만 표시하도록 결과를 필터링합니다. 이 쿼리는 리소스를 많이 소비할 수 있습니다.

예를 들어, 이 가이드의 [구체화된 뷰 생성](#) 섹션에 있는 mv_total_orders 구체화된 뷰를 살펴보세요. 이 뷰가 생성되면 Amazon Redshift는 자동으로 위의 쿼리를 다음과 같이 다시 작성합니다.

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

이 새 쿼리는 2개의 대형 테이블을 조인하는 대신 구체화된 뷰를 사용하기 때문에 훨씬 더 빠르게 실행됩니다.

쿼리를 다시 작성하여 다음과 같은 이유로 구체화된 뷰가 더 광범위하게 채택됩니다.

1. 구체화된 뷰는 투명한 쿼리 다시 작성을 지원합니다. 쿼리를 다시 작성하면 테이블 측면에서 표현된 SQL 문이 하나 이상의 구체화된 뷰에 액세스하는 문으로 변환됩니다. 이 변환은 최종 사용자에게 투명하므로 개입이 필요 없고 SQL 문에 구체화된 뷰에 대한 참조도 필요 없습니다.
2. 구체화된 뷰는 캐시된 공통 결과 세트의 적용을 간소화하고 사전 컴퓨팅과 같은 쿼리 간 최적화를 지원합니다.
3. 데이터 웨어하우스 및 데이터 통합 시나리오에서 구체화된 뷰는 여러 데이터 소스 간의 차이를 숨기기 위해 외부 테이블 결과를 구체화하여 로컬 복제본 또는 읽기/쓰기 분할을 구현할 수 있습니다.
4. 자동 구체화된 뷰 생성과 결합된 쿼리 다시 작성 기능은 데이터베이스 자율성을 가속화합니다.

구체화된 뷰의 장점

구체화된 뷰를 사용하면 다음과 같은 몇 가지 장점이 있습니다.

- 업데이트 횟수 감소 - 표준 뷰는 물리적으로 구체화되지 않습니다. 즉, 표준 뷰를 정의하는 쿼리는 쿼리에서 뷰를 참조할 때마다 실행됩니다. 반면, 구체화된 뷰는 일반 뷰처럼 미리 계산되어 쿼리 식의 결과로 디스크에 저장됩니다(객체와 유사). 표준 뷰와 달리 구체화된 뷰는 사용될 때마다 업데이트되지 않습니다.
- 응답 시간 단축 - 구체화된 뷰는 뷰보다 응답 속도가 빠릅니다. 이는 구체화된 뷰가 미리 계산되어 쿼리를 해결하거나 구체화된 뷰를 생성하는 쿼리에 조인하는 데 시간을 낭비하지 않기 때문입니다.
- 저장된 SQL 문 - 구체화된 뷰 대신 롤업 또는 집계 테이블을 사용할 수 있습니다. 롤업 테이블은 미리 계산되어 디스크에 저장되지만(구체화된 뷰와 유사), SQL 문을 데이터베이스에 저장하지는 않습니다. 구체화된 뷰는 해당 SQL 문을 저장합니다.
- 쉽게 새로 고침 - 구체화된 뷰는 새로 고치기 쉽습니다. REFRESH MATERIALIZED VIEW 명령을 실행하기만 하면 됩니다.
- 자동 쿼리 다시 작성 - 쿼리 최적화 프로그램은 구체화된 뷰가 SQL 문에 명시적으로 사용되지 않은 경우에도 기존 구체화된 뷰에서 데이터를 가져오기 위해 SQL 문을 다시 작성할 수 있습니다.

리소스

참조

- [Amazon Redshift에서 구체화된 뷰 생성](#)
- [구체화된 뷰 쿼리](#)
- [구체화된 뷰를 사용하기 위한 자동 쿼리 다시 작성](#)
- [구체화된 뷰 새로 고침](#)

도구

- [Amazon Redshift 쿼리 에디터 v2.0](#)

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 [RSS 피드](#)를 구독하십시오.

변경 사항	설명	날짜
최초 게시	—	2022년 12월 15일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.