



在 Amazon Redshift 中使用实体化视图

AWS 规范性指导



AWS 规范性指导: 在 Amazon Redshift 中使用实体化视图

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

简介 1

概览 1

目标业务成果 1

了解实体化视图 2

视图 2

实体化视图 2

视图类型比较 2

创建实体化视图 5

查询实体化视图 7

刷新实体化视图 8

自动查询重写 9

查询重写的工作原理 9

实体化视图的优点 11

资源 12

文档历史记录 13

..... xiv

在 Amazon Redshift 中使用实体化视图

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

2022 年 12 月 ([文档历史记录](#))

概览

数据仓库应用程序通常需要对大型表执行复杂的查询。处理这些查询可能既耗时又昂贵。例如，当您在 Amazon Redshift 中运行查询时，leader 节点会解析、验证、规划、优化和运行您的查询。此过程会占用大量挂钟时间、CPU 时间和内存，从而增加机会成本和实际成本。

本指南向您展示了如何使用 Amazon Redshift 中的实体化视图来加快查询速度，尤其是可预测且经常重复的查询。实体化视图通过存储预先计算的结果集来减少查询时间，这样就不必直接访问底层基表。反过来，减少查询时间可以降低查询处理的成本，从而以经济高效的方式扩展应用程序。本指南适用于数据工程师、数据架构师和数据分析师。

目标业务成果

您可以使用本指南实现以下业务成果：

- 降低处理 Amazon Redshift 查询的成本
- 经济高效地扩展您的应用程序
- 解放用户，让他们把时间花在价值更高的任务上

了解实体化视图

视图

视图是基于 SQL SELECT 查询结果集的虚拟表。虚拟表包含从查询表达式中检索的数据，但结果不存储在磁盘上。使用视图时，您获得 up-to-date 的数据总是最多，因为每次运行查询时，查询都会从原始表中提取数据。您可以从一个或多个基表或视图创建视图。您也可以像查询原始基表一样查询视图。

以下示例查询显示了如何创建视图：

```
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;
```

以下示例查询显示了如何查询视图：

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

实体化视图

实体化视图是包含查询结果的数据库对象。例如，实体化视图可以是以下任何一种：

- 远程数据的本地副本
- 表或联接结果的行或列的子集
- 使用聚合函数的摘要

视图类型比较

下表总结了视图和实体化视图之间的差异。

键	视图	实体化视图
定义	一种虚拟表，它不存储任何数据，而是运行定义的 SQL 查询，从数据库中的一个或多个表中获取数据	由可编辑 SQL 查询定义的虚拟表，但查询结果存储在磁盘上
存储	查询表达式的结果不存储在磁盘上，只有查询表达式存储在磁盘上	查询表达式和查询表达式的结果都存储在磁盘上
运行	每次在查询中引用视图时，定义视图的查询都会运行。	查询结果存储在磁盘上，每次用户尝试从实体化视图中获取数据时，查询表达式都不会运行
数据新近性	始终提供基表中的最新更新值	如果数据库中的值发生更改，则不提供最新更新的值
成本	无存储成本	有存储成本
设计	要创建标准视图，您必须： · 可访问基础表 · 使用标准 SELECT 语句	要创建实体化视图，您必须： · 可访问基础表 · 使用标准 SELECT 语句 或者，您可以指定： · 实体化视图是否包含在自动和手动集群快照中，这些快照存储在 Amazon Simple Storage Service (Amazon S3) 中 · 实体化视图中的数据是如何分布和排序的 · 实体化视图是否应根据基表中的最新更改自动刷新

使用量	不经常访问或更新数据时	经常访问或更新数据时
低使用量	本地数据库	本地数据库
中低使用量	本地数据库	本地数据库或云数据库
中高使用量	本地数据库或云数据库	云数据库
高使用量	云数据库	云数据库

不经常访问或更新数据时	经常访问或更新数据时
① 数据冗余度低，数据独立性高。 ② 数据冗余度低，数据冗余度低。 ③ 数据冗余度低，数据冗余度低。 ④ 数据冗余度低，数据冗余度低。 ⑤ 数据冗余度低，数据冗余度低。 ⑥ 数据冗余度低，数据冗余度低。 ⑦ 数据冗余度低，数据冗余度低。 ⑧ 数据冗余度低，数据冗余度低。 ⑨ 数据冗余度低，数据冗余度低。 ⑩ 数据冗余度低，数据冗余度低。	① 数据冗余度高，数据冗余度高。 ② 数据冗余度高，数据冗余度高。 ③ 数据冗余度高，数据冗余度高。 ④ 数据冗余度高，数据冗余度高。 ⑤ 数据冗余度高，数据冗余度高。 ⑥ 数据冗余度高，数据冗余度高。 ⑦ 数据冗余度高，数据冗余度高。 ⑧ 数据冗余度高，数据冗余度高。 ⑨ 数据冗余度高，数据冗余度高。 ⑩ 数据冗余度高，数据冗余度高。

经常访问或更新数据时

创建实体化视图

您可以基于以下一个或多个因素创建实体化视图：

- Amazon Redshift 表
- 您可以使用 [Amazon Redshift Spectrum](#) 或[联合查询](#)创建的外部表
- 其他实体化视图

在您创建实体化视图时，Amazon Redshift 会运行用户指定的 SQL 语句，以从基表或多个表中收集数据。然后，Amazon Redshift 存储结果集。

下图概述了名为 mv_total_orders 的实体化视图。此视图由 SQL 查询定义，该查询使用两个基表：客户和订单。

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

查询实体化视图

当您查询实体化视图时，您可以直接访问实体化视图中预先计算的数据。您可以通过引用实体化视图名称作为数据来源，在任何 SQL 查询中使用实体化视图，就像表或标准视图一样。

例如，请考虑本指南[创建实体化视图](#)部分中的 mv_total_orders 实体化视图示例插图。如果您想构建 mv_total_orders 查询（返回订单总额超过 500 美元的客户列表），则可以运行以下标准查询：

```
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;
```

但前面的查询并未针对速度进行优化。我们建议您改为运行以下查询：

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

推荐的查询运行速度要快得多，因为查询结果是预先计算的，并且不需要访问基础表（客户和订单）。Amazon Redshift 可以直接从 mv_total_orders 返回结果。

重要提示

当查询访问实体化视图时，查询只查看最近刷新时存储在实体化视图中的数据。因此，查询可能看不到实体化视图相应基表中的所有最新更改。

刷新实体化视图

实体化视图包含查询结果的快照。除非您将 Amazon Redshift 配置为定期更新，否则实体化视图不会定期更新。要手动刷新和更新实体化视图中的数据，您可以随时使用 `REFRESH MATERIALIZED VIEW` 语句。该命令可识别基表中发生的更改，并将这些更改应用到实体化视图中。

有两种刷新实体化视图的方法：手动刷新和自动刷新（称为[自动刷新](#)）。以下示例查询显示了如何手动刷新实体化视图：

```
REFRESH MATERIALIZED VIEW mv_total_orders;
```

要自动刷新实体化视图，请将 `AUTO REFRESH YES` 子句添加到 `CREATE MATERIALIZED VIEW` 语句，如以下示例所示：

```
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 会在基表更改后尽快自动刷新实体化视图。为了在处理刷新时最大限度地减少集群中活动工作负载的影响，Amazon Redshift 会考虑以下因素：

- 当前系统负载
- 刷新所需的资源
- 可用的集群资源
- 实体化视图的使用频率

Amazon Redshift 会优先处理您的工作负载，而不是自动刷新，并可能会停止自动刷新以保持用户工作负载的性能。请记住，这种方法可能会延迟实体化视图的刷新。对于刷新状态，您可以查看 [SVL_MV_REFRESH_STATUS](#) 视图。该视图记录用户发起或自动刷新的查询。

自动查询重写

Amazon Redshift 可以自动重写未明确引用用于提高性能的现有实体化视图的 SQL 查询。直接查询实体化视图可以减少查询处理时间，但必须修改查询语句。如果您使用实体化视图来加速查询，则必须考虑如何系统地和自动地使用实体化视图来响应查询。您可以使用透明重写来添加或删除实体化视图，方法与索引相同，不会影响现有 SQL 语句。Amazon Redshift 在自动重写查询时仅使用 up-to-date 物化视图。

查询重写的工作原理

考虑以下 SQL 查询。该查询联接两个表：客户（1000 多万行）和订单（100 多亿行）：

```
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;
```

该查询联接两个大表，对 o.amount 列应用 sum 聚合，然后筛选结果，仅显示订单超过 500 美元的客户。该查询可能会消耗大量资源。

例如，请考虑本指南[创建实体化视图](#)部分中的 mv_total_orders 实体化视图。创建此视图后，Amazon Redshift 会自动重写前面的查询，如下所示：

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

这个新查询的运行速度要快得多，因为它使用的是实体化视图，而不是联接两个大表。

由于以下原因，查询重写使得实体化视图得到更广泛的采用：

1. 实体化视图支持查询的透明重写。查询重写将用表来表示的 SQL 语句转换为访问一个或多个实体化视图的语句。这种转换对最终用户是透明的，无需干预，也无需在 SQL 语句中引用实体化视图。

2. 实体化视图简化了缓存通用结果集的应用，并支持跨查询优化，比如预计算。
3. 在数据仓库和数据集成场景中，实体化视图可以将外部表的结果实体化，以隐藏多个数据来源之间的差异，实现本地副本或读/写分离。
4. 查询重写功能与自动实体化视图相结合，加快了数据库自主性。

实体化视图的优点

使用实体化视图有几个优点：

- 更新更少 - 标准视图实际上不是实体化的，这意味着每当查询中引用视图时，系统都会运行定义标准视图的查询。相比之下，实体化视图是预先计算的并存储在磁盘（类似于对象）上，作为常规视图等查询表达式的结果。与标准视图不同，实体化视图不会在每次使用时更新。
- 更快的响应时间 - 与视图相比，实体化视图的响应速度更快。这是因为实体化视图是预先计算的，因此不会浪费时间解析查询或联接创建实体化视图的查询。
- 存储的 SQL 语句 - 您可以使用汇总表或聚合表来代替实体化视图。汇总表是预先计算的并存储在磁盘（类似于实体化视图）上，但它们不会将 SQL 语句存储在数据库中。实体化视图会存储其 SQL 语句。
- 易于刷新 - 实体化视图易于刷新。只需运行 `REFRESH MATERIALIZED VIEW` 命令。
- 自动重写查询 - 查询优化程序可以重写 SQL 语句，从现有的实体化视图中获取数据，即使实体化视图没有在 SQL 语句中明确使用。

资源

参考

- [在 Amazon Redshift 中创建实体化视图](#)
- [查询实体化视图](#)
- [自动查询重写以使用实体化视图](#)
- [刷新实体化视图](#)

工具

- [Amazon Redshift 查询编辑器 v2.0](#)

文档历史记录

下表介绍了本指南的一些重要更改。如果您希望收到有关未来更新的通知，可以订阅 [RSS 源](#)。

变更	说明	日期
初次发布	—	2022 年 12 月 15 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。