



Menggunakan tampilan terwujud di Amazon Redshift

AWS Panduan Preskriptif



AWS Panduan Preskriptif: Menggunakan tampilan terwujud di Amazon Redshift

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Pengantar	1
Gambaran Umum	1
Hasil bisnis yang ditargetkan	1
Memahami pandangan yang terwujud	2
Tampilan	2
Tampilan terwujud	2
Perbandingan jenis tampilan	3
Membuat tampilan terwujud	5
Menanyakan tampilan yang terwujud	7
Menyegarkan tampilan terwujud	9
Penulisan ulang kueri otomatis	11
Cara kerja penulisan ulang kueri	11
Keuntungan dari pandangan terwujud	13
Sumber daya	14
Riwayat dokumen	15
.....	xvi

Menggunakan tampilan terwujud di Amazon Redshift

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

Desember 2022 ([riwayat dokumen](#))

Gambaran Umum

Aplikasi gudang data sering mengharuskan Anda untuk melakukan kueri kompleks pada tabel besar. Memproses kueri ini bisa memakan waktu dan mahal. Misalnya, saat Anda menjalankan kueri di Amazon Redshift, node pemimpin mem-parsing, memvalidasi, merencanakan, mengoptimalkan, dan menjalankan kueri Anda. Proses ini dapat meningkatkan biaya peluang dan biaya aktual Anda dengan menggunakan sejumlah besar waktu jam dinding, waktu CPU, dan memori.

Panduan ini menunjukkan cara menggunakan tampilan terwujud di Amazon Redshift untuk mempercepat kueri, terutama kueri yang dapat diprediksi dan sering diulang. Tampilan terwujud mengurangi waktu kueri dengan menyimpan kumpulan hasil yang telah dihitung sebelumnya, sehingga Anda tidak perlu mengakses tabel dasar yang mendasarinya secara langsung. Pengurangan waktu kueri, pada gilirannya, dapat mengurangi biaya pemrosesan kueri sehingga Anda dapat menskalakan aplikasi Anda secara hemat biaya. Panduan ini ditujukan untuk insinyur data, arsitek data, dan analis data.

Hasil bisnis yang ditargetkan

Anda dapat menggunakan panduan ini untuk mencapai hasil bisnis berikut:

- Kurangi biaya pemrosesan kueri Amazon Redshift
- Kembangkan aplikasi Anda secara efisien dan hemat biaya
- Bebaskan pengguna untuk menghabiskan waktu pada tugas bernilai lebih tinggi

Memahami pandangan yang terwujud

Tampilan

Tampilan adalah tabel virtual yang didasarkan pada kumpulan hasil SELECT query SQL. Tabel virtual berisi data yang diambil dari ekspresi kueri, tetapi hasilnya tidak disimpan di disk. Saat Anda menggunakan tampilan, Anda selalu mendapatkan up-to-date data terbanyak karena kueri menarik data dari tabel asli setiap kali Anda menjalankan kueri. Anda dapat membuat tampilan dari satu atau beberapa tabel dasar atau tampilan. Anda juga dapat menanyakan tampilan dengan cara yang sama seperti Anda menanyakan tabel dasar asli.

Contoh query berikut menunjukkan cara membuat tampilan:

```
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;
```

Contoh query berikut menunjukkan bagaimana untuk query tampilan:

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

Tampilan terwujud

Tampilan terwujud adalah objek database yang berisi hasil kueri. Misalnya, tampilan terwujud dapat berupa salah satu dari yang berikut:

- Salinan data lokal yang terletak dari jarak jauh
- Sebuah subset dari baris atau kolom dari tabel atau hasil gabungan
- Ringkasan menggunakan fungsi agregat

Perbandingan jenis tampilan

Tabel berikut merangkum perbedaan antara tampilan dan tampilan terwujud.

Kunci	Lihat	Tampilan terwujud
Definisi	Tabel virtual yang tidak menyimpan data apa pun, tetapi menjalankan kueri SQL yang ditentukan untuk mendapatkan data dari satu atau beberapa tabel dalam database	Tabel virtual yang didefinisikan oleh kueri SQL yang dapat diedit, tetapi hasil kueri disimpan di disk
Penyimpanan	Hasil ekspresi kueri tidak disimpan di disk — hanya ekspresi kueri yang disimpan di disk	Ekspresi kueri dan hasil dari ekspresi kueri keduanya disimpan pada disk
Jalankan .	Kueri yang mendefinisikan tampilan berjalan setiap kali tampilan direferensikan dalam kueri	Hasil kueri disimpan di disk, dan ekspresi kueri tidak berjalan setiap kali pengguna mencoba mengambil data dari tampilan yang terwujud
Kebaruan data	Selalu berikan nilai terbaru yang diperbarui dari tabel dasar	Tidak memberikan nilai terbaru yang diperbarui jika nilai itu diubah dalam database
Biaya	Tidak ada biaya penyimpanan	Memiliki biaya penyimpanan
Desain	Untuk membuat tampilan standar, Anda harus: · Memiliki akses ke tabel yang mendasarinya · Gunakan SELECT pernyataan standar	Untuk membuat tampilan terwujud, Anda harus: · Memiliki akses ke tabel yang mendasarinya · Gunakan SELECT pernyataan standar

Secara opsional, Anda dapat menentukan yang berikut:

- Apakah tampilan terwujud disertakan dalam snapshot cluster otomatis dan manual, yang disimpan di Amazon Simple Storage Service (Amazon S3)
- Bagaimana data dalam tampilan terwujud didistribusikan dan diurutkan
- Apakah tampilan yang terwujud harus disegarkan secara otomatis dengan perubahan terbaru dari tabel dasarnya

Penggunaan

Ketika data diakses atau diperbarui jarang

Ketika data sering diakses atau diperbarui

Membuat tampilan terwujud

Anda dapat membuat tampilan terwujud berdasarkan satu atau beberapa hal berikut:

- Tabel Amazon Redshift
- Tabel eksternal yang dapat Anda buat menggunakan [Amazon Redshift Spectrum](#) atau kueri [gabungan](#)
- Pandangan terwujud lainnya

Saat Anda membuat tampilan terwujud, Amazon Redshift menjalankan pernyataan SQL yang ditentukan pengguna untuk mengumpulkan data dari tabel dasar atau tabel. Kemudian, Amazon Redshift menyimpan set hasil.

Ilustrasi berikut memberikan gambaran umum tentang tampilan terwujud yang disebut.

`mv_total_orders` Tampilan ini ditentukan oleh kueri SQL yang menggunakan dua tabel dasar: pelanggan dan pesanan.

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

Menanyakan tampilan yang terwujud

Saat Anda menanyakan tampilan terwujud, Anda langsung mengakses data yang telah dihitung sebelumnya dalam tampilan terwujud. Anda dapat menggunakan tampilan terwujud dalam kueri SQL apa pun dengan mereferensikan nama tampilan terwujud sebagai sumber data, seperti dalam tabel atau tampilan standar.

Misalnya, pertimbangkan ilustrasi contoh tampilan `mv_total_orders` terwujud dari bagian [Membuat tampilan terwujud](#) dari panduan ini. Jika Anda ingin membuat kueri untuk `mv_total_orders` (yang mengembalikan daftar pelanggan yang memiliki pesanan dengan total lebih dari \$500), maka Anda dapat menjalankan kueri standar berikut:

```
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;
```

Namun, kueri sebelumnya tidak dioptimalkan untuk kecepatan. Kami menyarankan Anda menjalankan kueri berikut sebagai gantinya:

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

Kueri yang direkomendasikan berjalan lebih cepat karena hasil kueri sudah dihitung sebelumnya, dan tidak perlu mengakses tabel yang mendasarinya (pelanggan dan pesanan). Amazon Redshift dapat mengembalikan hasilnya langsung dari `mv_total_orders`

 Penting

Saat kueri mengakses tampilan terwujud, kueri hanya melihat data yang disimpan dalam tampilan terwujud pada penyegaran terbarunya. Oleh karena itu, kueri mungkin tidak melihat semua perubahan terbaru dari tabel dasar yang sesuai dari tampilan terwujud.

Menyegarkan tampilan terwujud

Tampilan terwujud berisi snapshot dari hasil kueri. Tampilan terwujud tidak diperbarui secara berkala, kecuali jika Anda mengonfigurasi Amazon Redshift untuk melakukan pembaruan berkala. Untuk menyegarkan dan memperbarui data secara manual dalam tampilan terwujud, Anda dapat menggunakan `REFRESH MATERIALIZED VIEW` pernyataan kapan saja. Perintah ini mengidentifikasi perubahan yang terjadi di tabel dasar dan menerapkan perubahan tersebut ke tampilan terwujud.

Ada dua cara untuk menyegarkan tampilan yang terwujud: penyegaran manual dan penyegaran otomatis (disebut [autorefreshing](#)). Contoh kueri berikut menunjukkan cara menyegarkan tampilan terwujud secara manual:

```
REFRESH MATERIALIZED VIEW mv_total_orders;
```

Untuk mengautorefresh tampilan terwujud, tambahkan `AUTO REFRESH YES` klausa ke `CREATE MATERIALIZED VIEW` pernyataan seperti yang ditunjukkan oleh contoh berikut:

```
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 secara otomatis menyegarkan tampilan terwujud sesegera mungkin setelah tabel dasar berubah. Untuk meminimalkan dampak beban kerja aktif di klaster saat memproses penyegaran, Amazon Redshift mempertimbangkan faktor-faktor berikut:

- Beban sistem saat ini
- Sumber daya yang dibutuhkan untuk penyegaran
- Sumber daya cluster yang tersedia
- Seberapa sering pandangan terwujud digunakan

Amazon Redshift memprioritaskan beban kerja Anda daripada autorefresh dan dapat menghentikan autorefreshing untuk mempertahankan kinerja beban kerja pengguna. Perlu diingat bahwa pendekatan ini dapat menunda penyegaran beberapa pandangan yang terwujud. Untuk status penyegaran, Anda dapat memeriksa tampilan [SVL_MV_REFRESH_STATUS](#). Tampilan ini mencatat kueri yang diprakarsai pengguna atau diautorefreshed.

Penulisan ulang kueri otomatis

Amazon Redshift dapat secara otomatis menulis ulang kueri SQL yang tidak secara eksplisit mereferensikan tampilan terwujud yang ada yang digunakan untuk peningkatan kinerja. Menanyakan tampilan terwujud secara langsung mengurangi waktu pemrosesan kueri, tetapi pernyataan kueri harus dimodifikasi. Jika Anda menggunakan tampilan terwujud untuk mempercepat kueri, penting untuk mempertimbangkan cara menggunakan tampilan terwujud secara sistematis dan otomatis untuk menanggapi kueri. Anda dapat menggunakan penulisan ulang transparan untuk menambah atau menghapus tampilan terwujud dengan cara yang sama seperti indeks, tanpa memengaruhi pernyataan SQL yang ada. Amazon Redshift hanya menggunakan tampilan up-to-date terwujud saat secara otomatis menulis ulang kueri.

Cara kerja penulisan ulang kueri

Pertimbangkan query SQL berikut. Kueri menggabungkan dua tabel: pelanggan (10+ juta baris) dan pesanan (10+ miliar baris):

```
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;
```

Kueri ini menggabungkan dua tabel besar, menerapkan agregasi jumlah pada `o.amount` kolom, dan kemudian memfilter hasilnya untuk menampilkan hanya pelanggan yang memesan lebih dari \$500. Kueri ini bisa menghabiskan banyak sumber daya.

Sebagai contoh, pertimbangkan tampilan `mv_total_orders` terwujud dari bagian [Membuat tampilan terwujud](#) dari panduan ini. Setelah tampilan ini dibuat, Amazon Redshift secara otomatis menulis ulang kueri sebelumnya sebagai berikut:

```
SELECT cust_id,  
       first_name,  
       total_amount
```

```
FROM mv_total_orders  
WHERE total_amount > 500;
```

Kueri baru ini berjalan lebih cepat karena menggunakan tampilan terwujud alih-alih menggabungkan dua tabel besar.

Penulisan ulang kueri telah menyebabkan adopsi yang lebih luas dari pandangan terwujud karena alasan berikut:

1. Tampilan terwujud mendukung penulisan ulang kueri yang transparan. Penulisan ulang kueri mengubah pernyataan SQL yang dinyatakan dalam tabel menjadi pernyataan yang mengakses satu atau lebih tampilan terwujud. Transformasi ini transparan bagi pengguna akhir, tidak memerlukan intervensi dan tidak ada referensi ke tampilan terwujud dalam pernyataan SQL.
2. Tampilan terwujud menyederhanakan penerapan set hasil umum yang di-cache dan mendukung pengoptimalan lintas kueri seperti precomputing.
3. Dalam gudang data dan skenario integrasi data, tampilan terwujud dapat mewujudkan hasil tabel asing untuk menyembunyikan perbedaan di antara beberapa sumber data, menerapkan replika lokal atau pemisahan baca/tulis.
4. Fitur penulisan ulang kueri yang dikombinasikan dengan pembuatan tampilan terwujud otomatis mempercepat otonomi basis data.

Keuntungan dari pandangan terwujud

Ada beberapa keuntungan menggunakan tampilan terwujud:

- Lebih sedikit pembaruan — Tampilan standar tidak terwujud secara fisik, yang berarti bahwa kueri yang mendefinisikan tampilan standar berjalan setiap kali tampilan direferensikan dalam kueri. Sebaliknya, tampilan terwujud dihitung sebelumnya dan disimpan pada disk (mirip dengan objek) sebagai hasil dari ekspresi kueri seperti tampilan reguler. Tidak seperti tampilan standar, tampilan terwujud tidak diperbarui setiap kali digunakan.
- Waktu respons yang lebih cepat — Tampilan yang terwujud merespons lebih cepat dibandingkan dengan tampilan. Ini karena tampilan terwujud telah dihitung sebelumnya dan oleh karena itu tidak membuang waktu untuk menyelesaikan kueri atau bergabung dalam kueri yang membuat tampilan terwujud.
- Pernyataan SQL tersimpan - Anda dapat menggunakan tabel rollup, atau agregasi, alih-alih tampilan terwujud. Tabel rollup dihitung sebelumnya dan disimpan pada disk (mirip dengan tampilan terwujud), tetapi mereka tidak menyimpan pernyataan SQL mereka dalam database. Tampilan terwujud menyimpan pernyataan SQL mereka.
- Mudah disegarkan - Tampilan terwujud mudah disegarkan. Cukup jalankan `REFRESH MATERIALIZED VIEW` perintah.
- Penulisan ulang kueri otomatis — Pengoptimal kueri dapat menulis ulang pernyataan SQL Anda untuk mengambil data dari tampilan terwujud yang ada, bahkan jika tampilan terwujud tidak digunakan secara eksplisit dalam pernyataan SQL Anda.

Sumber daya

Referensi

- [Membuat tampilan terwujud di Amazon Redshift](#)
- [Menanyakan tampilan yang terwujud](#)
- [Penulisan ulang kueri otomatis untuk menggunakan tampilan terwujud](#)
- [Menyegarkan tampilan yang terwujud](#)

Alat

- [Editor Kueri Pergeseran Merah Amazon v2.0](#)

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan [RSS](#).

Perubahan	Deskripsi	Tanggal
Publikasi awal	—	15 Desember 2022

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.