Rakamin Academy

id/x partners

Project Based Internship

Data Transformation & Query Scheduling

Optimizing Performance and Efficiency: Exploring Data Transformation and Query Scheduling in Data Warehouses

Daftar Isi

A. Introduction

Data warehouse adalah proses pengumpulan, penyimpanan, dan pengelolaan data dalam jumlah besar untuk mendukung analisis dan pengambilan keputusan. Transformasi data adalah proses mengubah data mentah menjadi format yang sesuai untuk analisis, pelaporan, dan kueri. Query Scheduling adalah proses pengoptimalan eksekusi kueri kompleks di data warehouse untuk meningkatkan kinerja dan mengurangi konsumsi sumber daya. Pada artikel ini, kita akan mengeksplorasi pentingnya transformasi data dan query scheduling dalam data warehouse, dan bagaimana mereka membantu untuk membuat keputusan yang tepat berdasarkan data yang akurat dan relevan.

B. Data Transformation

Transformasi data adalah langkah penting dalam proses data warehouse karena membantu memastikan bahwa data yang disimpan di data warehouse bersih, akurat, dan berguna untuk analisis. Dengan mengubah data mentah menjadi format yang terstruktur dan terorganisir, kueri dan analisis data menjadi lebih mudah, yang pada gilirannya menghasilkan wawasan dan keputusan yang lebih akurat.

Selain itu, transformasi data membantu memastikan bahwa data dari sumber yang berbeda dapat digabungkan dan dianalisis bersama, bahkan jika data tersebut dalam format yang berbeda atau memiliki ketidakkonsistenan. Hal ini sangat penting untuk organisasi yang menggunakan data dari berbagai sumber, karena mungkin sulit untuk menggabungkan data dari berbagai sumber tanpa terlebih dahulu mengubahnya menjadi format umum. Berikut beberapa contoh

dan penerapan yang dilakukan dalam proses transformasi data pada data warehouse dengan SQL.

1. **Cleaning Data**

   Salah satu kasus penggunaan umum untuk transformasi data adalah membersihkan data. Ini melibatkan penghapusan semua ketidakkonsistenan atau kesalahan dalam data agar lebih mudah digunakan. Misalnya, jika database berisi data pelanggan dengan nilai yang tidak konsisten untuk "negara" (seperti "CA", "california", dan "CA"), kita dapat menggunakan SQL untuk mengubah data menjadi format standar (misal " CA") untuk memudahkan analisis.

```
UPDATE customers
SET state = 'CA'
WHERE state IN ('california', 'CA.')
```

2. **Agregasi Data**

   Kasus penggunaan umum lainnya untuk transformasi data adalah agregasi data. Ini melibatkan penggabungan data dari beberapa baris atau tabel menjadi satu baris atau tabel. Misalnya, jika database berisi data penjualan untuk perusahaan dengan banyak produk, kita dapat menggunakan SQL untuk mengubah data tersebut menjadi tabel ringkasan yang menampilkan total penjualan untuk setiap produk.

```
SELECT product_name, SUM(sales) as total_sales
FROM sales_data
GROUP BY product_name
```

3. **Join**

Join adalah cara lain untuk transformasi data menggunakan SQL. Ini melibatkan penggabungan data dari dua atau lebih tabel. Misalnya, jika kita memiliki database dengan data pelanggan dan data pesanan, kita dapat menggunakan SQL untuk menggabungkan dua tabel menjadi satu tabel yang menunjukkan data pelanggan dan pesanan.

```
SELECT *
FROM customers
JOIN orders ON customers.id = orders.customer_id
```

4. **Normalization**

Normalisasi adalah proses mengubah data menjadi format standar yang dioptimalkan untuk analisis. Misalnya, jika database berisi data tentang beberapa produk dan atributnya (seperti ukuran, warna, dan harga), kita dapat menggunakan SQL untuk mengubah data menjadi tabel terpisah untuk setiap atribut, yang memudahkan analisis dan perbandingan data di seluruh produk.

```
-- Create a table for product sizes
CREATE TABLE product_sizes (
  id INT PRIMARY KEY,
  product_id INT,
  size VARCHAR(10)
);
-- Insert data into product_sizes
INSERT INTO product_sizes (id, product_id, size)
SELECT id, product_id, size
```

```
FROM products;
-- Remove size column from products table
ALTER TABLE products DROP COLUMN size;
```

C. Query Scheduling

Query scheduling adalah fitur yang memungkinkan Anda mengotomatiskan eksekusi kueri SQL di lingkungan data warehouse. Query scheduling memungkinkan Anda untuk mengotomatiskan tugas berulang, seperti ekstraksi data dan pemuatan data, dan membantu Anda memastikan bahwa data terbaru dan tersedia saat Anda membutuhkannya. Query scheduling menjadi penting karena beberapa alasan, antara lain:

- Kesegaran Data: query scheduling membantu Anda memastikan bahwa data terbaru dan tersedia saat Anda membutuhkannya. Ini penting untuk lingkungan data warehouse di mana data diperbarui secara teratur.
- Otomasi: query scheduling mengotomatiskan tugas berulang, seperti ekstraksi data dan pemuatan data, yang menghemat waktu dan mengurangi risiko kesalahan manusia.
- Performa: query scheduling memungkinkan Anda menjadwalkan kueri di luar jam sibuk, yang dapat meningkatkan performa data warehouse Anda.

Query scheduling berfungsi dengan menjadwalkan kueri SQL untuk dijalankan pada interval yang ditentukan. Kueri terjadwal dapat dijalankan secara berulang atau pada waktu tertentu. Query scheduling biasanya dikelola oleh alat penjadwalan, seperti sistem manajemen database (DBMS) atau penjadwal pekerjaan seperti windows scheduler maupun crontab scheduler. Saat kueri

terjadwal dijalankan, kueri dijalankan di server data warehouse, dan hasilnya dikembalikan ke klien. Hasil kueri dapat disimpan dalam tabel, dikirim ke file, atau digunakan oleh aplikasi lain.

Berikut adalah contoh cara menyiapkan penjadwalan kueri di lingkungan SQL data warehouse:

1. Tentukan frekuensi kueri: Kueri harus dijalankan setiap hari.
2. Tulis kueri SQL: Kueri SQL berikut mengekstrak data penjualan dari tabel dan menyimpannya dalam file:

```
SELECT *
FROM Sales
WHERE order_date = CURRENT_DATE;
```

3. Konfigurasikan alat penjadwalan: Di SQL data warehouse, Anda dapat menggunakan tools scheduler pada DBMS yang digunakan untuk menyiapkan penjadwalan kueri. Untuk menyiapkan penjadwalan kueri, Anda akan membuat alur di tools scheduler yang menjalankan kueri SQL dan menampilkan hasilnya ke file. Anda kemudian akan menjadwalkan pipeline untuk berjalan setiap hari.
4. Uji jadwal: Setelah jadwal disiapkan, Anda akan menguji jadwal untuk memastikan bahwa kueri berjalan dengan benar dan bahwa data diekstraksi dan disimpan dengan benar.

REFERENCE

https://www.upsolver.com/resources/guides/sql-data-transformation

https://towardsdatascience.com/cleaning-and-transforming-data-with-sql-f93c4de0d2fc

https://www.ibm.com/docs/en/spectrum-protect/8.1.9?topic=commands-query-schedule-query-schedules