



Project Based Internship

Aggregation & Normalization

Making Sense of Your Data Warehouse: A Comprehensive Guide to Aggregation and Normalization



Daftar Isi

A. Introduction	3
B. Aggregation	3
C. Normalization	5
Reference	9

A. Introduction

Dalam dunia data warehouse, istilah "agregasi" dan "normalisasi" sering digunakan untuk menjelaskan teknik pengorganisasian dan pemrosesan data dalam jumlah besar. Dalam SQL, bahasa yang digunakan untuk mengelola database relasional, agregasi dan normalisasi adalah konsep kunci yang memungkinkan analisis dan pelaporan data yang efisien. Agregasi mengacu pada proses meringkas kumpulan data besar dengan mengelompokkannya bersama-sama dan menghitung statistik seperti rata-rata, jumlah, dan hitungan. Teknik ini umumnya digunakan dalam intelijen dan analitik bisnis untuk mengekstrak wawasan dari sejumlah besar data. Normalisasi, di sisi lain, adalah proses penataan tabel sedemikian rupa sehingga mengurangi redundansi data dan meningkatkan integritas data. Teknik ini melibatkan penguraian data menjadi tabel yang lebih kecil dan lebih mudah dikelola serta membangun hubungan di antara mereka. Normalisasi memastikan bahwa data konsisten, akurat, dan mudah ditanyakan. Pada artikel ini, kita akan mengeksplorasi bagaimana agregasi dan normalisasi bekerja dalam konteks data warehouse menggunakan SQL. Artikel akan memberikan contoh bagaimana teknik ini dapat digunakan untuk mengekstrak insight dari kumpulan data besar dan meningkatkan efisiensi pemrosesan data.

B. Aggregation

Agregasi adalah proses meringkas data di satu atau lebih dimensi. Pada data warehouse, agregasi biasanya melibatkan pengambilan kumpulan data besar dan mengurangnya menjadi kumpulan data agregat yang lebih kecil, yang dapat dianalisis dan ditafsirkan dengan lebih mudah. Misalnya, Anda dapat mengambil tabel data penjualan dengan jutaan baris, dan meringkasnya

berdasarkan bulan, wilayah, dan kategori produk untuk lebih memahami bisnis Anda.

- **SQL Function For Data Aggregation**

SQL menyediakan berbagai fungsi yang dapat digunakan untuk melakukan operasi agregasi pada data Anda. Berikut adalah beberapa yang paling umum:

1. **SUM:** Menghitung jumlah kumpulan nilai dalam kolom.
2. **AVG:** Menghitung rata-rata sekumpulan nilai dalam kolom.
3. **COUNT:** Menghitung jumlah baris dalam tabel atau jumlah nilai bukan nol dalam kolom.
4. **MIN:** Menemukan nilai minimum dalam kolom.
5. **MAX:** Menemukan nilai maksimum dalam kolom.
6. **GROUP BY:** Mengelompokkan data menurut satu atau beberapa kolom, lalu menjalankan fungsi agregasi pada setiap grup.

Mari kita lihat beberapa contoh bagaimana fungsi ini dapat digunakan dalam SQL.

Contoh 1: Mencari Total Penjualan

Misalkan kita memiliki tabel sales dengan kolom berikut: Order ID, Customer ID, Product, Quantity, dan Price. Kita dapat menggunakan fungsi SUM untuk menemukan total penjualan:

```
SELECT SUM(Quantity * Price) AS TotalSales  
FROM SalesTable;
```

Query diatas akan memberi output data total penjualan di semua pesanan.

Contoh 2: Mencari Harga Rata-Rata

Kita dapat menggunakan fungsi AVG untuk menemukan harga rata-rata semua produk:

```
SELECT AVG(Price) AS AveragePrice  
FROM SalesTable;
```

Query diatas akan memberi output data harga rata-rata di semua produk.

Contoh 3: Pengelompokan berdasarkan Kategori Produk

Kita dapat menggunakan fungsi GROUP BY untuk mengelompokkan data penjualan berdasarkan kategori produk, lalu mencari total penjualan untuk setiap kategori:

```
SELECT ProductCategory, SUM(Quantity * Price) AS TotalSales  
FROM SalesTable  
GROUP BY ProductCategory;
```

Query diatas akan memberi output data total penjualan untuk setiap kategori produk.

C. Normalization

Normalisasi data adalah teknik yang digunakan dalam data warehouse untuk meningkatkan integritas, konsistensi, dan kegunaan data. Normalisasi mengurangi redundansi data, menghilangkan ketidakkonsistenan dan redundansi, serta memudahkan kueri dan analisis data. Normalisasi data adalah proses mengatur tabel data dalam basis data relasional sehingga memenuhi persyaratan tertentu. Persyaratan ini meliputi:

- Menghilangkan data yang redundan

- Memastikan konsistensi data
- Membuat data mudah untuk kueri
- Meningkatkan integritas data

Normalisasi diperlukan karena data yang tidak ditata dengan benar dapat menyebabkan inkonsistensi data, kesalahan, dan kueri yang lambat. Normalisasi meningkatkan akurasi data dan membuat pergudangan data lebih efisien. Ada beberapa level normalisasi yang dapat dipenuhi oleh tabel database, antara lain:

1. First Normal Form (1NF)

Bentuk normal pertama mengharuskan setiap kolom dalam tabel hanya berisi nilai atomik. Ini berarti bahwa setiap sel dalam kolom hanya boleh menampung satu informasi. Tabel yang tidak memenuhi persyaratan ini dianggap tidak dinormalisasi.

kode_faktur	tanggal	kode_barang	nama_barang	harga	qty
KD_001	13/07/2020	BRG_001	Indomie Goreng	2500	5
		BRG_002	Indomie Goreng Jumbo	3000	8
		BRG_003	Mie Sedaap Goreng	2500	9
		BRG_004	Mie Sedaap Soto	2300	3
KD_002	13/07/2020	BRG_005	Intermie Goreng	1500	6
		BRG_006	Intermie Soto	1500	6
KD_003	14/07/2020	BRG_007	Pop Mie Ayam	4500	3

Gambar 1 Data yang Belum Dinormalisasi

Tabel yang belum dinormalisasi ini, jika dinormalisasi maka bentuk normalisasi pertamanya (1NF) adalah:

1NF					
kode_faktur	tanggal	kode_barang	nama_barang	harga	qty
KD_001	13/07/2020	BRG_001	Indomie Goreng	2500	5
KD_001	13/07/2020	BRG_002	Indomie Goreng Jumbo	3000	8
KD_001	13/07/2020	BRG_003	Mie Sedaap Goreng	2500	9
KD_001	13/07/2020	BRG_004	Mie Sedaap Soto	2300	3
KD_002	13/07/2020	BRG_005	Intermie Goreng	1500	6
KD_002	13/07/2020	BRG_006	Intermie Soto	1500	6
KD_003	14/07/2020	BRG_007	Pop Mie Ayam	4500	3

Gambar 2 Tabel Normalisasi 1NF

2. Second Normal Form (2NF)

Bentuk normal kedua dibangun di atas bentuk normal pertama dengan mensyaratkan bahwa setiap kolom yang bukan kunci (non-key) dalam sebuah tabel bergantung pada primary keynya. Ini berarti bahwa setiap kolom yang bukan kunci (non-key) harus dikaitkan dengan primary key dengan cara yang unik dan tidak berlebihan. Intinya adalah pada tahap normalisasi 2NF ini tabel tersebut harus dipecah berdasarkan primary key. Sehingga bentuk normalisasi 2NF dari tabel tersebut adalah sebagai berikut:

2NF						
Tabel Barang			Tabel Transaksi			
kode_barang	nama_barang	harga	kode_faktur	tanggal	kode_barang	qty
BRG_001	Indomie Goreng	2500	KD_001	13/07/2020	BRG_001	5
BRG_002	Indomie Goreng Jumbo	3000	KD_001	13/07/2020	BRG_002	8
BRG_003	Mie Sedaap Goreng	2500	KD_001	13/07/2020	BRG_003	9
BRG_004	Mie Sedaap Soto	2300	KD_001	13/07/2020	BRG_004	3
BRG_005	Indomie Goreng	1500	KD_002	13/07/2020	BRG_005	6
BRG_006	Indomie Soto	1500	KD_002	13/07/2020	BRG_006	6
BRG_007	Pop Mie Ayam	4500	KD_003	14/07/2020	BRG_007	3

Gambar 3 Tabel Normalisasi 2NF

3. Third Normal Form (3NF)

Bentuk normal ketiga dibangun di atas bentuk normal kedua dengan mensyaratkan bahwa setiap kolom bukan kunci (non-key) dalam sebuah tabel tidak bergantung pada kolom bukan kunci (non-key) lainnya. Ini berarti bahwa setiap kolom bukan kunci (non-key) hanya boleh berisi informasi yang terkait dengan kunci utama. Contohnya ada pada atribut qty, kolom tersebut tidak bergantung langsung pada primary key kode_faktur melainkan bergantung pada kolom kode_barang. Jadi setelah dinormalisasi 3NF akan menghasilkan tabel berikut:

3NF					
Tabel Barang			Tabel Transaksi		
kode_barang	nama_barang	harga	kode_faktur	tanggal	
BRG_001	Indomie Goreng	2500	KD_001	13/07/2020	
BRG_002	Indomie Goreng Jumbo	3000	KD_002	13/07/2020	
BRG_003	Mie Sedaap Goreng	2500	KD_003	14/07/2020	
BRG_004	Mie Sedaap Soto	2300			
BRG_005	Intermie Goreng	1500			
BRG_006	Intermie Soto	1500			
BRG_007	Pop Mie Ayam	4500			
Tabel Detail Barang					
kode_faktur	kode_barang	qty	harga		
KD_001	BRG_001	5	2500		
KD_001	BRG_002	8	2500		
KD_001	BRG_003	9	2500		
KD_001	BRG_004	3	2500		
KD_002	BRG_005	6	3000		
KD_002	BRG_006	6	3000		
KD_003	BRG_007	3	2500		

Gambar 4 Tabel Normalisasi 3NF

4. Fourth Normal Form (4NF)

Bentuk normal keempat dibangun di atas bentuk normal ketiga dengan mensyaratkan bahwa tidak ada beberapa kumpulan data terkait dalam sebuah tabel. Artinya, sebuah tabel tidak boleh berisi lebih dari satu kumpulan nilai yang terkait dengan kunci utama (primary key).

5. Fifth Normal Form (5NF)

Bentuk normal kelima dibangun di atas bentuk normal keempat dengan mengharuskan tidak ada lagi kunci komposit dalam sebuah tabel. Ini berarti bahwa sebuah tabel tidak boleh mengandung lebih banyak kunci yang memiliki lebih dari satu atribut.



Reference

<https://aantamim.id/normalisasi-database/>

<https://www.simplilearn.com/tutorials/sql-tutorial/sql-aggregate-functions>

<https://www.simplilearn.com/tutorials/sql-tutorial/what-is-normalization-in-sql>

