

Soal & Template Jawaban

Task 5

Nama : Makmurry Akbar

Petunjuk

Silahkan merujuk pada Data Source Task 5 yang telah disediakan untuk mengerjakan soal soal di bawah ini

Pada bagian data analytics, terdiri dari 4 soal dengan use case & tabel yang sama. Bayangkan kamu memiliki database erp yang terdiri dari 3 tabel: penjualan, pelanggan, barang. Tabel tersebut akan dibuat menjadi sebuah datamart yang nantinya digunakan untuk visualisasi.

Query

Soal 1 *:

Dari 2 query ini, mana yang bekerja lebih baik? Jelaskan mengapa.

- a) `SELECT * FROM pelanggan WHERE SUBSTR(alamat, 1, 3) = Mat;`
- b) `SELECT * FROM pelanggan WHERE alamat LIKE 'Mat%'`

*disclaimer: soal ini tidak terkait dengan data source

Jawaban : b

Alasan : Dengan operator logika LIKE diikuti dengan wildcard % setelah teks 'Mat' akan mencocokkan nilai yang mengandung teks 'Mat' dengan 'M' huruf besar dan 'at' huruf kecil karena operator LIKE peka terhadap huruf besar dan kecil, lalu diikuti dengan teks apapun setelah teks 'Mat'. Misalnya nama awalan alamat adalah 'Mataram' maka akan muncul nilai Mataram diikuti dengan teks selanjutnya.

Jika kita memilih jawaban a , maka query akan error karena sintaks SUBSTR tidak perlu ditambah operator sama dengan (=)

Query

Soal 2 *:

Anggap kita memiliki tabel pelanggan dengan kolom: id, nama, tanggal_lahir, alamat. Bagaimana cara yang lebih tepat dalam menulis query untuk mendapatkan data pelanggan yang tanggal_lahir nya ada di antara 2000-01-01 sampai 2008-12-31? Pilihlah salah satu jawaban dan berikan alasannya.

- a) `SELECT * FROM pelanggan WHERE tanggal_lahir >= '2000-01-01' AND tanggal_lahir <= '2008-12-31'`
- b) `SELECT * FROM pelanggan WHERE tanggal_lahir BETWEEN '2000-01-01' AND '2008-12-31'`

*disclaimer: soal ini tidak terkait dengan data source

Jawaban : b

Alasan : sebenarnya kedua query akan mengembalikan hasil yang sama , saya memilih query b dengan operator BETWEEN karena penulisan syntaxnya lebih singkat jadi tidak perlu menulis nama kolom dua kali dan waktu eksekusi query sedikit lebih cepat dibanding operator perbandingan.

Soal 3: Menentukan Primary Key

A. Tugas

Tentukan primary key dari table penjualan. jelaskan alasannya

B. Jawaban & Penjelasan : id_invoice

Penjelasan : karena setiap baris id_invoice mewakili 1 transaksi dan berisi nilai unik.

Soal 4: Design Datamart

A. Tugas

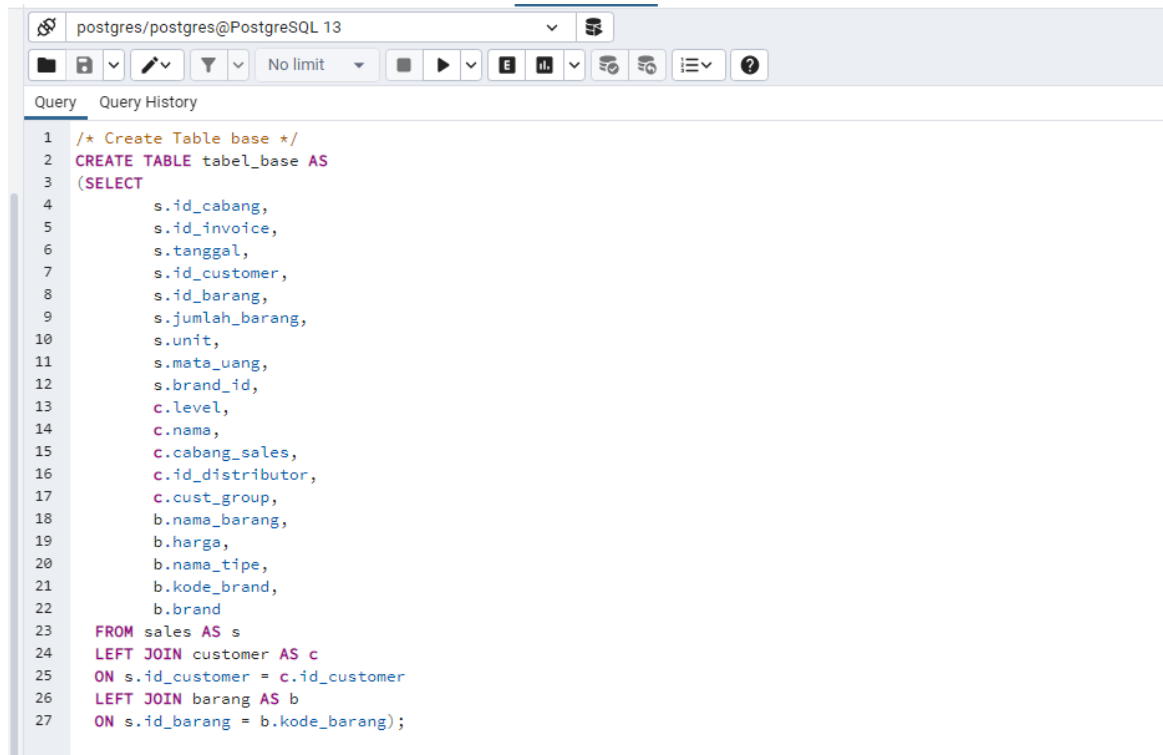
Buatlah design datamart (Terdiri dari tabel base, dan tabel aggregate). Upload file query dalam gdrivemu (pastikan dapat diakses public). Lalu masukkan linknya di tabel di bawah, dan cantumkan juga screenshoot query nya (jika lebih dari 1 file, maka masing masing file di-screenshoot)

Silahkan tambah halaman jika dibutuhkan

B. Jawaban :

No	Nama File	Link
1	tabel_base.sql	https://drive.google.com/file/d/1aoCxB6nqwPm4fpG0aeEhx0DAf1L7Jyb/view?usp=share_link
2	tabel_aggregate.sql	https://drive.google.com/file/d/13u-f2mHaV6wUc_MFypsLW6was0hRtgfH/view?usp=share_link

Table Base "<<tabel_base>>"



The screenshot shows a PostgreSQL query editor window titled 'postgres/postgres@PostgreSQL 13'. The editor contains a SQL query to create a table named 'tabel_base' by selecting data from three existing tables: 'sales', 'customer', and 'barang'. The query is as follows:

```
1  /* Create Table base */
2  CREATE TABLE tabel_base AS
3  (SELECT
4      s.id_cabang,
5      s.id_invoice,
6      s.tanggal,
7      s.id_customer,
8      s.id_barang,
9      s.jumlah_barang,
10     s.unit,
11     s.mata_uang,
12     s.brand_id,
13     c.level,
14     c.nama,
15     c.cabang_sales,
16     c.id_distributor,
17     c.cust_group,
18     b.nama_barang,
19     b.harga,
20     b.nama_tipe,
21     b.kode_brand,
22     b.brand
23  FROM sales AS s
24  LEFT JOIN customer AS c
25  ON s.id_customer = c.id_customer
26  LEFT JOIN barang AS b
27  ON s.id_barang = b.kode_barang);
```

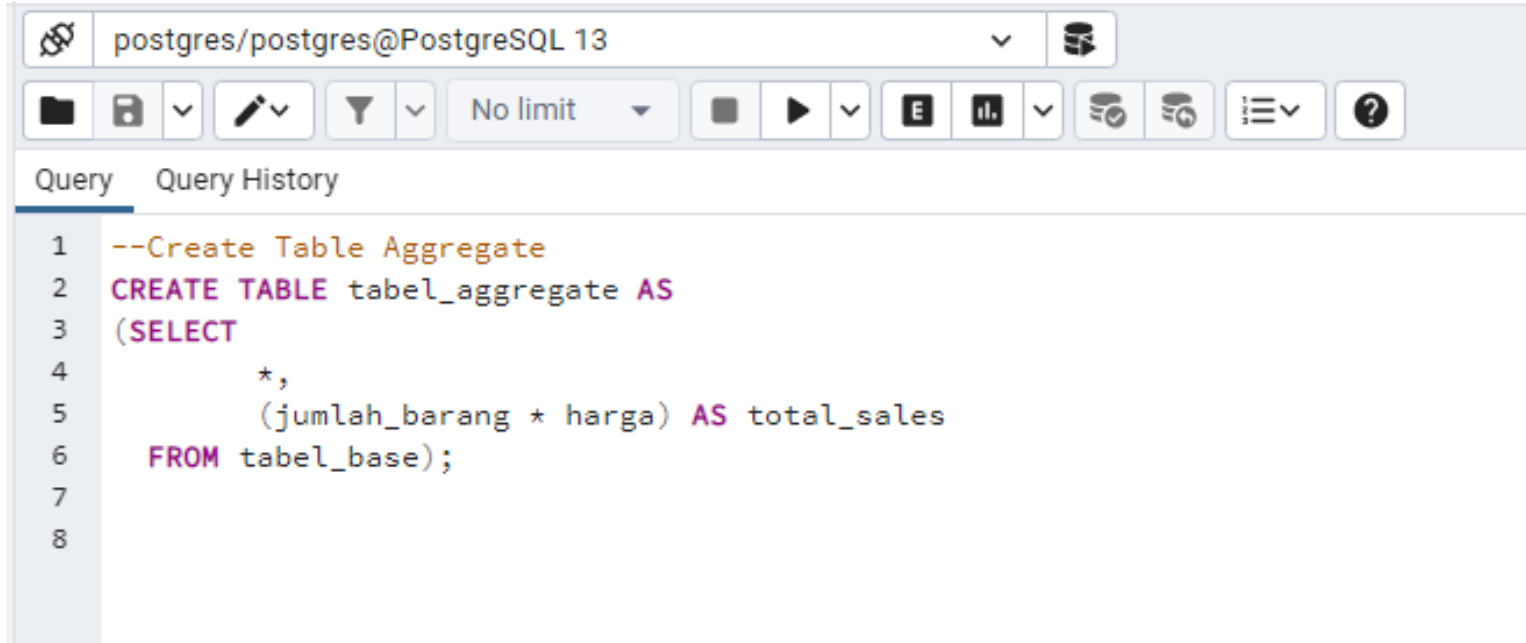
Table Base “<<tabel_base>>”

column	data type	description	transformation
Id_cabang	VARCHAR	ID setiap cabang	
id_invoice	VARCHAR	ID dari invoice penjualan Salicyl	
tanggal	DATE	Tanggal transaksi	
id_customer	VARCHAR	ID pelanggan	
id_barang	VARCHAR	ID barang	
jumlah_barang	INT	Jumlah barang per transaksi	
unit	VARCHAR	Bentuk kemasan	
mata_uang	VARCHAR	Mata uang yang digunakan	
brand_id	VARCHAR	ID setiap brand	

Table Base “<<tabel_base>>”

level	VARCHAR	Level pelanggan	
nama	VARCHAR	Nama pelanggan	
cabang_sales	VARCHAR	Kota setiap cabang	
id_distributor	VARCHAR	ID distributor	
cust_group	VARCHAR	Kategori Pelanggan	Kolom group menjadi cust_group
nama_barang	VARCHAR	Nama barang	
harga	INT	Harga barang	
nama_tipe	VARCHAR	Kategori hasil produk	
kode_brand	INT	Kode brand	
brand	VARCHAR	Nama brand	

Table Aggregate “<<tabel_aggregate>>”



The screenshot shows a PostgreSQL query editor interface. At the top, the connection is set to 'postgres/postgres@PostgreSQL 13'. Below the connection bar is a toolbar with various icons for file operations, query execution, and settings. The main area displays a SQL query in a monospaced font, with line numbers 1 through 8 on the left. The query is a SQL standard 'CREATE TABLE AS' statement that creates a table named 'tabel_aggregate' by selecting all columns from 'tabel_base' and calculating a new column 'total_sales' as the product of 'jumlah_barang' and 'harga'.

```
1  --Create Table Aggregate
2  CREATE TABLE tabel_aggregate AS
3  (SELECT
4      *,
5      (jumlah_barang * harga) AS total_sales
6  FROM tabel_base);
7
8
```

Table Aggregate “<<tabel_aggregate>>”

column	data type	description	transformation
Id_cabang	VARCHAR	ID setiap cabang	
id_invoice	VARCHAR	ID dari invoice penjualan Salicyl	
tanggal	DATE	Tanggal transaksi	
id_customer	VARCHAR	ID pelanggan	
id_barang	VARCHAR	ID barang	
jumlah_barang	INT	Jumlah barang per transaksi	
unit	VARCHAR	Bentuk kemasan	
mata_uang	VARCHAR	Mata uang yang digunakan	

Table Aggregate “<<tabel_aggregate>>”

brand_id	VARCHAR	ID setiap brand	
level	VARCHAR	Level pelanggan	
nama	VARCHAR	Nama pelanggan	
cabang_sales	VARCHAR	Kota setiap cabang	
id_distributor	VARCHAR	ID distributor	
cust_group	VARCHAR	Kategori Pelanggan	Kolom group menjadi cust_group
nama_barang	VARCHAR	Nama barang	
harga	INT	Harga barang	
nama_tipe	VARCHAR	Kategori hasil produk	

Table Aggregate “<<tabel_aggregate>>”

kode_brand	INT	Kode brand	
brand	VARCHAR	Nama brand	
total_sales	INT	Total penjualan	(jumlah_barang * harga) AS total_sales

Soal 5 : Data Visualization

A. Tugas

buatlah data visualiasasi nya, dan cantumkan linknya di bawah (pastikan bisa diakses publik).Lalu cantumkan juga screenshot visualisasinya

Silahkan tambah halaman jika dibutuhkan

B. Jawaban :

Link visualisasi :

<https://datastudio.google.com/reporting/b765fe96-37e2-43c4-9a91-ada384450701>

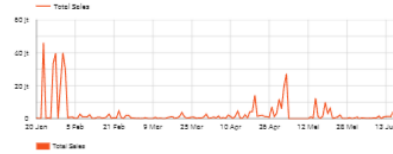


KIMIA FARMA SALES DASHBOARD 2022

Tanggal

Bulan

Sales Trend

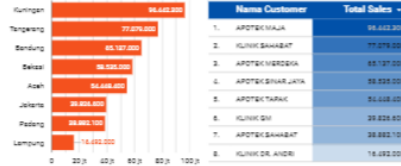


Total Sales
446.842.400

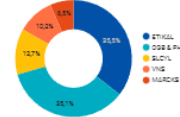
Rata-Rata Unit Penjualan
26

Pilih Filter

- Cabang Sales
- Nama Customer
- Brand
- Nama Barang



Sales By Brand



Nama Barang	Total Sales
1. ADICLOVIRUS	133.360.000
2. ALERGIE TABLET SAUT	116.890.000
3. ERGOTAMINE COFFEE	10.801.000
4. KLOPROFENAZIN TABLET SAUT	39.080.000
5. KETOCOANAZOLE TABLET 200 MG	33.308.000
6. TRAMADOL KAPSL 50 MG	24.632.000
7. AMFIDOLIN	22.728.000
8. AMERDOLIN MC	21.111.000
9. PARACETAMOL	17.283.000
10. TETRACICLINE KAPSL 250 MG	8.270.000

1-8/8

Sales By City



Soal 6 : Additional Complementary Data

A. Tugas :

Dari data yang tersedia, menurut kamu untuk melengkapi analisis nya apakah diperlukan data lain juga?
jika iya, sebutkan data apa yang kamu maksud dan mengapa memerlukan data tersebut

B. Jawaban :

1. id_customer seharusnya mewakili 1 nama pelanggan tapi dalam data pelanggan, satu nama customer memiliki banyak id_customer sehingga terjadi ambigu antara id_invoice dan id_customer dalam mewakili satu transaksi karena primary key data pelanggan seharusnya id_customer bukan nama.
2. Data profit , untuk mengetahui tren profit perusahaan
3. Data target profit, target penjualan untuk menentukan KPI