

HELLO THERE!

I'm Fajar Kurnia, and I'll be sharing with you
my project in Virtual Intern Rakamin X
KimiaFarma.



X





PROJECT OVERVIEW

Petunjuk

Silahkan merujuk pada Data Source yang telah disediakan untuk membuat Visualisasi

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.

Design Datamart

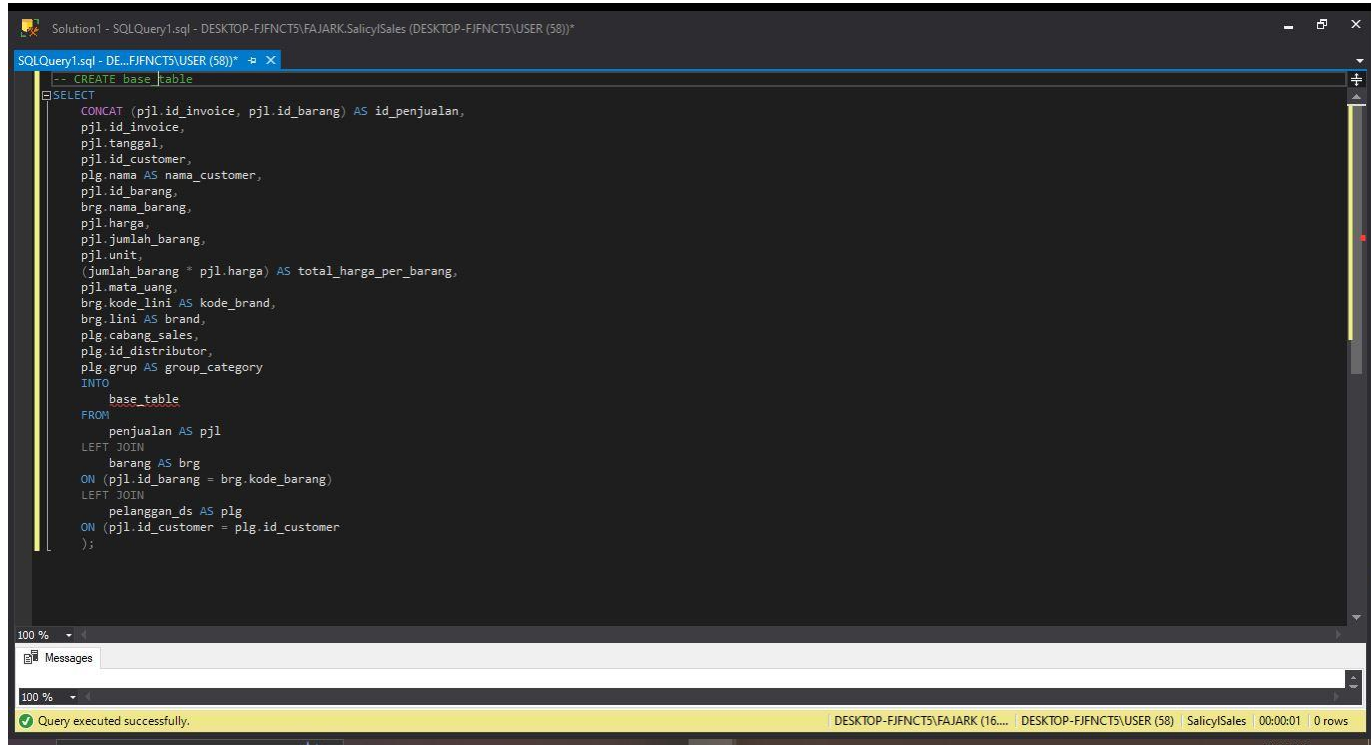
A. Tugas

Buatlah design datamart (Terdiri dari tabel base, dan tabel aggregate). Upload file query dalam gdrive mu (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)

A. Jawaban : Tool yang dipakai SSMS

No	Nama File	Link
1.	Create_database	https://drive.google.com/file/d/1NntgwJmOK-ysWHqJMBu4dUBkmmvpVPdB/view?usp=sharing
2.	base_table	https://drive.google.com/file/d/19AWCeVVHGUATukbX-Eu-XsgFzphBBDp3/view?usp=sharing
3.	aggregate_table	https://drive.google.com/file/d/17VmmRm8y4ZIZJHyHJY5YHlbpsQBekSan/view?usp=sharing

Table Base “base_table”



```
Solution1 - SQLQuery1.sql - DESKTOP-FJFNCT5\FAJARK\Sality\Sales (DESKTOP-FJFNCT5\USER (58))*
```

```
-- CREATE base table
SELECT
    CONCAT (pjl.id_invoice, pjl.id_barang) AS id_penjualan,
    pjl.id_invoice,
    pjl.tanggal,
    pjl.id_customer,
    plg.nama AS nama_customer,
    pjl.id_barang,
    brg.nama_barang,
    pjl.harga,
    pjl.jumlah_barang,
    pjl.unit,
    (jumlah_barang * pjl.harga) AS total_harga_per_barang,
    pjl.mata_uang,
    brg.kode_lini AS kode_brand,
    brg.lini AS brand,
    plg.cabang_sales,
    plg.id_distributor,
    plg.grup AS group_category
INTO
    base_table
FROM
    penjualan AS pjl
LEFT JOIN
    barang AS brg
ON (pjl.id_barang = brg.kode_barang)
LEFT JOIN
    pelanggan_ds AS plg
ON (pjl.id_customer = plg.id_customer
);
```

100 %

Messages

100 %

Query executed successfully. | DESKTOP-FJFNCT5\FAJARK (16... | DESKTOP-FJFNCT5\USER (58) | Sality\Sales | 00:00:01 | 0 rows

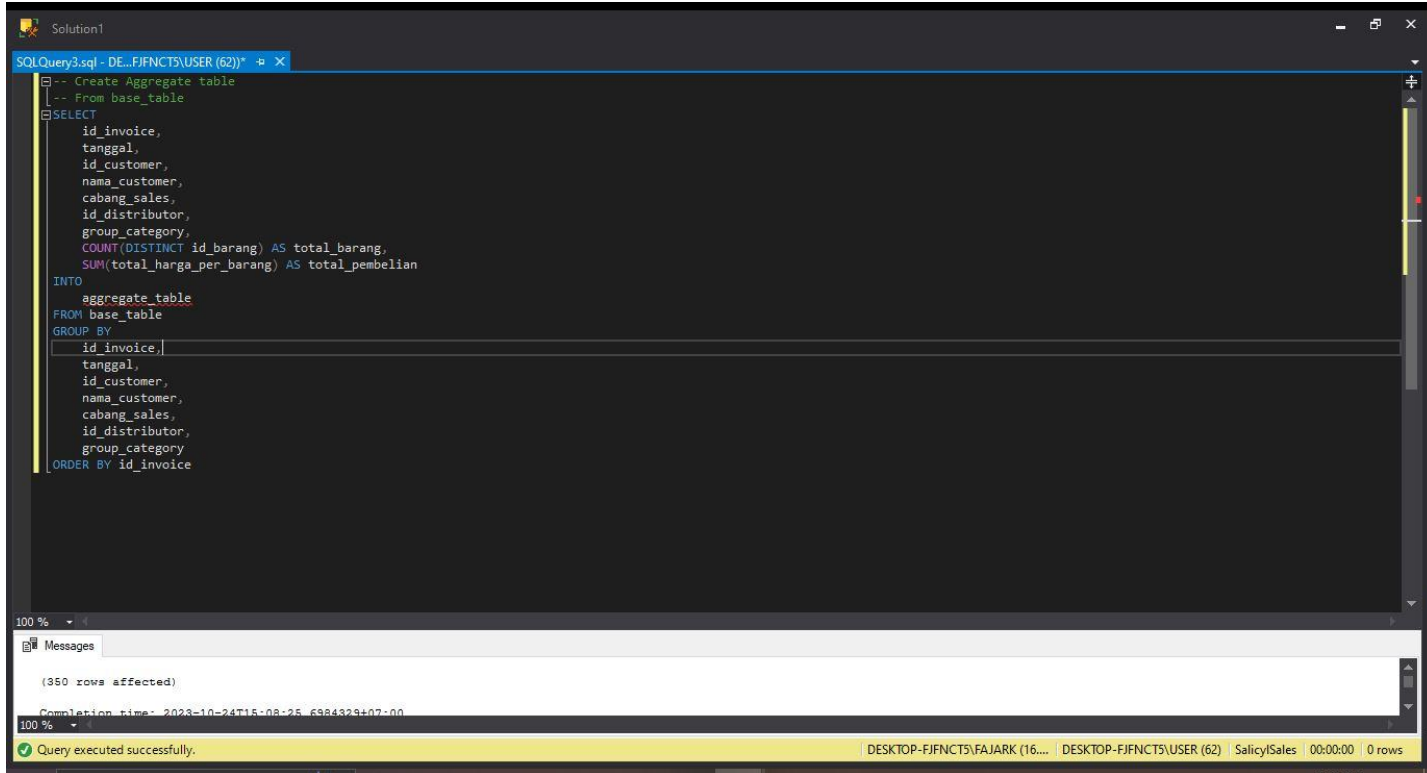
Table Base “base_table”

column	data type	description	transformation
Id_penjualan	varchar(13)	primary key base_table	CONCAT(pj.id_invoice, pj.id_barang) AS id_penjualan,
id_invoice	varchar(6)	id invoice	
tanggal	date	tanggal transaksi	
id_customer	varchar(9)	id customer, key join ke tabel pelanggan	
nama_customer	varchar(20)	Nama customer	nama AS nama_customer
id_barang	varchar(7)	id barang, key join ke tabel barang	
nama_barang	varchar(41)	nama barang	
harga	float	Harga barang	
jumlah_barang	int	jumlah barang yang dibeli	

Table Base “base_table”

column	data type	description	transformation
unit	varchar(3)	tipe kemasan setiap penjualan	
total_harga_per_barang	float	Jumlah harga per barang	jumlah_barang*harga
mata_uang	varchar(3)	mata uang penjualan dalam IDR	
kode_brand	int	kode brand	kode_lini AS kode_brand
brand	varchar(8)	nama brand	
cabang_sales	varchar(50)	kota cabang sales	
id_distributor	varchar(8)	id distributor	
group_category	varchar(10)	Kategori took (klinik/apotek)	grup AS group_category

Table Aggregate “aggregate_table”



The screenshot shows a SQL query editor window titled "Solution1" with a file named "SQLQuery3.sql". The query is written in T-SQL and is designed to create an aggregate table. The query starts with a comment "-- Create Aggregate table" and another comment "-- From base_table". The main query is a SELECT statement that selects various columns from "base_table" and calculates two aggregate functions: "COUNT(DISTINCT id_barang) AS total_barang" and "SUM(total_harga_per_barang) AS total_pembelian". The results are stored in a table named "aggregate_table". The query is executed, and the status bar at the bottom indicates that the query was executed successfully, affecting 350 rows.

```
-- Create Aggregate table
-- From base_table
SELECT
    id_invoice,
    tanggal,
    id_customer,
    nama_customer,
    cabang_sales,
    id_distributor,
    group_category,
    COUNT(DISTINCT id_barang) AS total_barang,
    SUM(total_harga_per_barang) AS total_pembelian
INTO
    aggregate_table
FROM base_table
GROUP BY
    id_invoice,
    tanggal,
    id_customer,
    nama_customer,
    cabang_sales,
    id_distributor,
    group_category
ORDER BY id_invoice
```

100 %
Messages
(350 rows affected)
Completion time: 2023-10-24T15:08:25.6984329+07:00
100 %
Query executed successfully.

DESKTOP-FJFNCT5\FAJARK (16... | DESKTOP-FJFNCT5\USER (62) | SalicytSales | 00:00:00 | 0 rows

Table Aggregate “aggregate_table”

column	data type	description	transformation
id_penjualan	varchar(13)	primary key aggregate table	
id_invoice	varchar(6)	id invoice	
tanggal	date	tanggal transaksi	
id_customer	varchar(9)	id customer, key join ke tabel pelanggan	
nama_customer	varchar(20)	nama customer	
nama_barang	varchar(41)	nama barang	
cabang_sales	varchar(50)	kota cabang sales	
id_distributor	varchar(8)	id distributor	
group_category	varchar(10)	Kategori took (klinik/apotek)	
jumlah_barang	int	jumlah barang yang dibeli	

Table Aggregate “aggregate_table”

column	data type	description	transformation
total_barang	int	total_barang	COUNT(DISTINCT id_barang) AS total_barang
total_pembelian	float	total transaksi	SUM(total_harga_per_barang) AS total_pembelian



DATA VISUALIZATION

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

A. Jawaban :

Link visualisasi (Google Data Studio) : <https://lookerstudio.google.com/reporting/bcc82458-1159-40c0-8652-60b2186b4306>

Dashboard

