

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 : Query B bekerja lebih baik dari pada Query A

Alasan : Karena tidak efisien menggunakan fungsi pada column yang difilter. Apabila column alamat tersebut di index, index yang tujuannya untuk mengoptimalkan query jadi tidak berguna karena ada fungsi di dalam column nya.

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 : Query B lebih baik dari pada Query A

Alasan : karena lebih baik kita menghindari operator AND, OR, & NOT. Apabila column tanggal_lahir tersebut di index. index yang tujuannya untuk mengoptimalkan query jadi tidak berguna karena ada operator tersebut. Sehingga lebih baik kita menggunakan cara lain sebagai penggantinya. Dalam case ini, AND digantikan dengan BETWEEN

Soal 3: Menentukan Primary Key

A. Tugas

Tentukan primary key dari table penjualan. jelaskan alasannya

B. Jawaban & Penjelasan : Pada table penjualan, tidak ada column yang unique sehingga primary key nya perlu di define sendiri.

Caranya dengan menggabungkan id_voice dan id_barang

id_penjualan	id_distributor	id_cabang	id_invoice
IN5997_BRG0001	TD	CAB01	IN5997
IN6190_BRG0001	TD	CAB10	IN6190
IN6113_BRG0001	TD	CAB09	IN6113
IN6173_BRG0001	EPM	CAB01	IN6173
IN6321_BRG0001	EPM	CAB04	IN6321
IN6028_BRG0001	EPM	CAB03	IN6028
IN6306_BRG0001	EPM	CAB09	IN6306
IN6219_BRG0001	TA	CAB01	IN6219
IN6241_BRG0001	EPM	CAB04	IN6241
IN6233_BRG0001	EPM	CAB03	IN6233

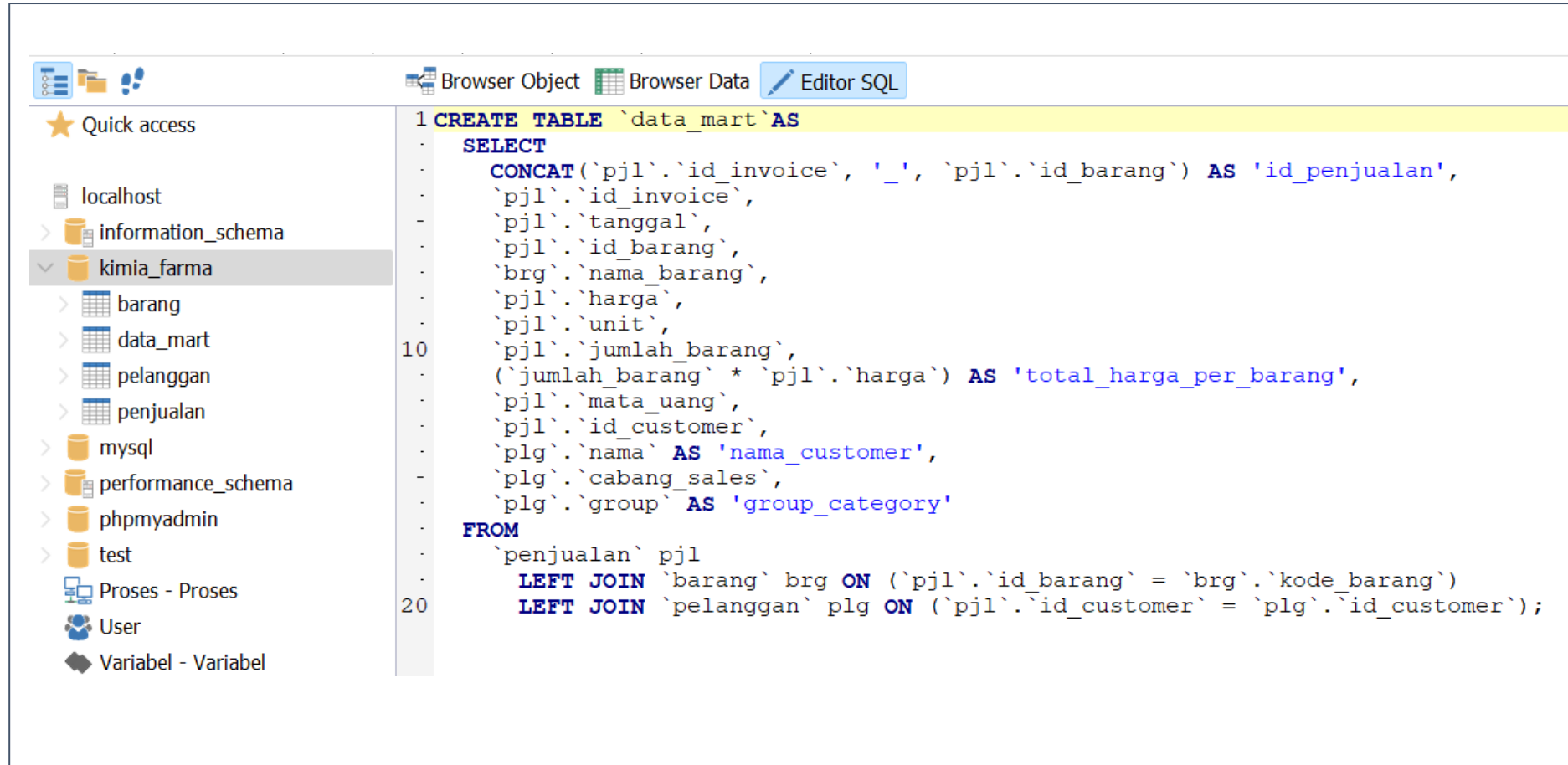
```
CREATE TABLE data_mart AS
SELECT
    concat(pjl.id_invoice, '_', pjl.id_barang) as id_penjualan,
```

Soal 4: 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)

Table Base "<< Penjualan >>"



The screenshot displays a database management interface with a sidebar on the left and a main editor area on the right. The sidebar, titled "Quick access", lists various database components: "localhost", "information_schema", "kimia_farma" (expanded), "barang", "data_mart", "pelanggan", "penjualan", "mysql", "performance_schema", "phpmyadmin", "test", "Proses - Proses", "User", and "Variabel - Variabel". The main editor area has tabs for "Browser Object", "Browser Data", and "Editor SQL". The "Editor SQL" tab is active, showing the following SQL code:

```
1 CREATE TABLE `data_mart` AS
2 SELECT
3     CONCAT(`pjl`.`id_invoice`, '_', `pjl`.`id_barang`) AS 'id_penjualan',
4     `pjl`.`id_invoice`,
5     `pjl`.`tanggal`,
6     `pjl`.`id_barang`,
7     `brg`.`nama_barang`,
8     `pjl`.`harga`,
9     `pjl`.`unit`,
10    `pjl`.`jumlah_barang`,
11    (`jumlah_barang` * `pjl`.`harga`) AS 'total_harga_per_barang',
12    `pjl`.`mata_uang`,
13    `pjl`.`id_customer`,
14    `plg`.`nama` AS 'nama_customer',
15    `plg`.`cabang_sales`,
16    `plg`.`group` AS 'group_category'
17 FROM
18     `penjualan` pjl
19     LEFT JOIN `barang` brg ON (`pjl`.`id_barang` = `brg`.`kode_barang`)
20     LEFT JOIN `pelanggan` plg ON (`pjl`.`id_customer` = `plg`.`id_customer`);
```

Table Base “<< Data Mart >>”

column	data type	description	transformation
Id_penjualan	String	PK dari table penjualan	CONCAT (id_invoice, '_' . Id_barang)
Id_invoice	String	Id invoice	-
Tanggal	Date	Tanggal transaksi dilakukan	-
Id_barang	String	Id barang, key join ke tabel barang	-
Nama_barang	String	Nama barang	-
Harga	Numeric	Harga barang	-
Unit	String	Kemasan untuk setiap penjualan	-

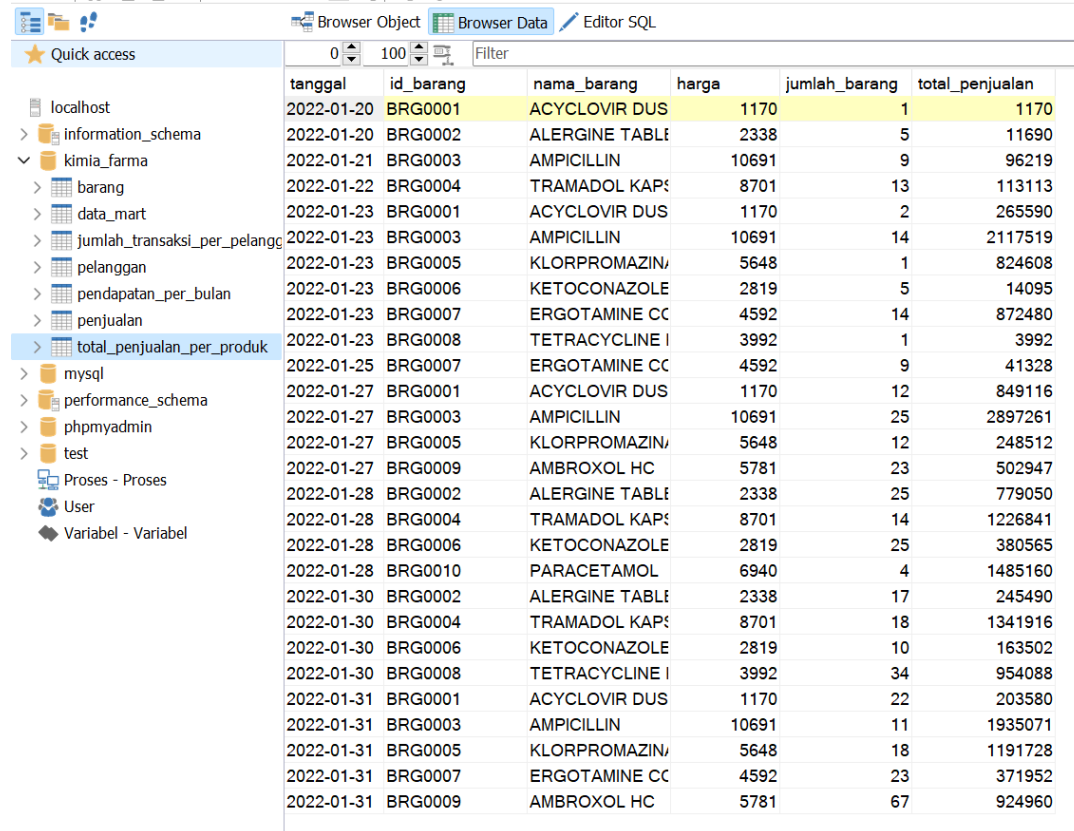
Table Base “<< Data Mart >>”

column	data type	description	transformation
Jumlah_barang	Numeric	Jumlah barang yang diambil	-
Total_harga_per_barang	Numeric	Harga barang x jumlah barang	(brg.jumlah_barang * pjl.harga)
Mata_uang	String	Mata uang. Default = IDR	-
Id_customer	String	Id customer, key join ke tabel pelanggan	-
Nama_customer	String	Nama pelanggan	-
Cabang_sales	String	Cabang yang supply barang ke pelanggan	-
Group_category	String	Kategori pelanggan, possible value: - Apotek - Klinik	-

Table Aggregate “<< Total_penjualan_per_produk >>”

```
1 -- Total penjualan per produk
CREATE TABLE `total_penjualan_per_produk` AS
SELECT
    DATE_FORMAT(`tanggal`, '%Y-%m-%d') AS 'tanggal',
    `id_barang`,
    `nama_barang`,
    `unit`,
    `harga`,
    `jumlah_barang`,
10 SUM(`harga` * `jumlah_barang`) AS 'total_penjualan'
FROM
    `data_mart`
WHERE
    DATE_FORMAT(`tanggal`, '%Y-%m-%d') BETWEEN '2022-01-01' AND '2022-01-31'
GROUP BY
    `tanggal`, `id_barang`;
```

Table Aggregate “<< Total_penjualan_per_produk >>”



The screenshot shows a database management interface with a sidebar on the left and a main data table on the right. The sidebar contains a tree view of database objects, including 'localhost', 'information_schema', 'kimia_farma', 'barang', 'data_mart', 'jumlah_transaksi_per_pelanggan', 'pelanggan', 'pendapatan_per_bulan', 'penjualan', 'total_penjualan_per_produk' (highlighted), 'mysql', 'performance_schema', 'phpmyadmin', 'test', 'Proses - Proses', 'User', and 'Variabel - Variabel'. The main table displays data for the 'total_penjualan_per_produk' table, with columns: tanggal, id_barang, nama_barang, harga, jumlah_barang, and total_penjualan. The table contains 24 rows of data, showing sales transactions for various products over time.

tanggal	id_barang	nama_barang	harga	jumlah_barang	total_penjualan
2022-01-20	BRG0001	ACYCLOVIR DUS	1170	1	1170
2022-01-20	BRG0002	ALERGINE TABL	2338	5	11690
2022-01-21	BRG0003	AMPICILLIN	10691	9	96219
2022-01-22	BRG0004	TRAMADOL KAP	8701	13	113113
2022-01-23	BRG0001	ACYCLOVIR DUS	1170	2	265590
2022-01-23	BRG0003	AMPICILLIN	10691	14	2117519
2022-01-23	BRG0005	KLORPROMAZIN	5648	1	824608
2022-01-23	BRG0006	KETOCONAZOLE	2819	5	14095
2022-01-23	BRG0007	ERGOTAMINE CC	4592	14	872480
2022-01-23	BRG0008	TETRACYCLINE I	3992	1	3992
2022-01-25	BRG0007	ERGOTAMINE CC	4592	9	41328
2022-01-27	BRG0001	ACYCLOVIR DUS	1170	12	849116
2022-01-27	BRG0003	AMPICILLIN	10691	25	2897261
2022-01-27	BRG0005	KLORPROMAZIN	5648	12	248512
2022-01-27	BRG0009	AMBROXOL HC	5781	23	502947
2022-01-28	BRG0002	ALERGINE TABL	2338	25	779050
2022-01-28	BRG0004	TRAMADOL KAP	8701	14	1226841
2022-01-28	BRG0006	KETOCONAZOLE	2819	25	380565
2022-01-28	BRG0010	PARACETAMOL	6940	4	1485160
2022-01-30	BRG0002	ALERGINE TABL	2338	17	245490
2022-01-30	BRG0004	TRAMADOL KAP	8701	18	1341916
2022-01-30	BRG0006	KETOCONAZOLE	2819	10	163502
2022-01-30	BRG0008	TETRACYCLINE I	3992	34	954088
2022-01-31	BRG0001	ACYCLOVIR DUS	1170	22	203580
2022-01-31	BRG0003	AMPICILLIN	10691	11	1935071
2022-01-31	BRG0005	KLORPROMAZIN	5648	18	1191728
2022-01-31	BRG0007	ERGOTAMINE CC	4592	23	371952
2022-01-31	BRG0009	AMBROXOL HC	5781	67	924960

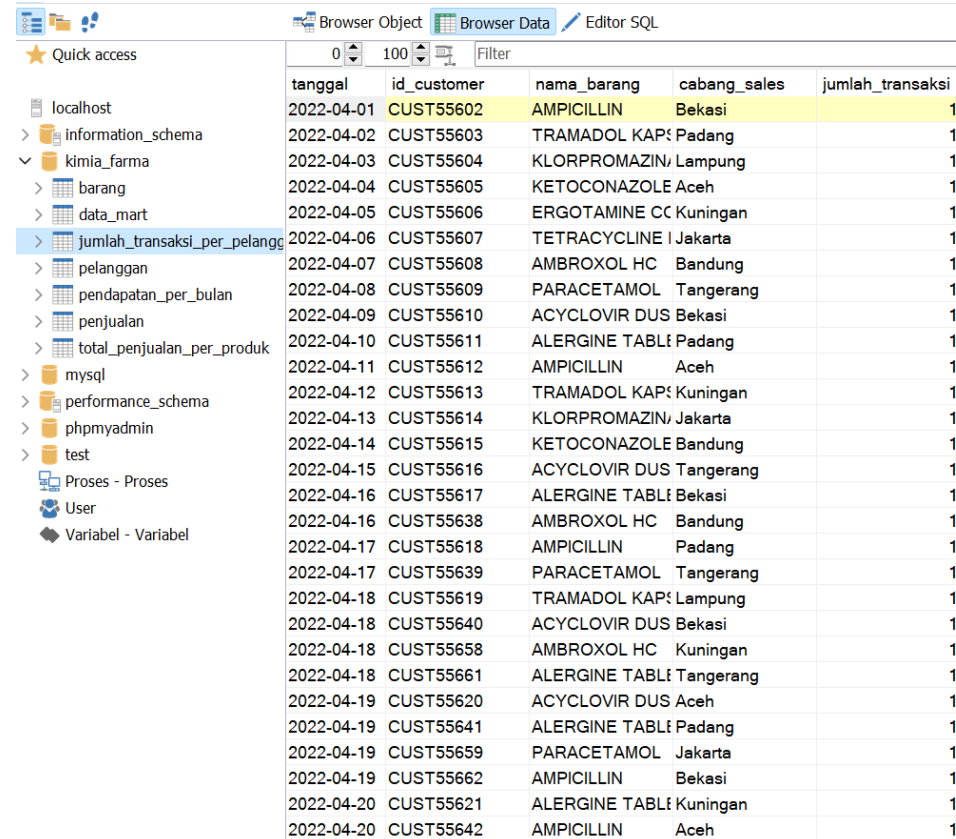
Table Aggregate “<< Total_penjualan_per_produk >>”

column	data type	description	transformation
Tanggal	Date	Tanggal transaksi dilakukan	DATE FORMAT (tanggal, %y-%m-%d)
Id_barang	String	Id barang, key join ke tabel barang	-
Nama_barang	String	Nama barang	-
Harga	Numeric	Harga barang	-
Jumlah_barang	Numeric	Jumlah barang yang diambil	-
Total_penjualan	Numeric	Total penjualan produk	SUM (harga * jumlah_barang)

Table Aggregate “<< Jumlah_Transaksi_per_pelanggan >>”

```
.  
20 -- Jumlah transaksi per pelanggan  
. CREATE TABLE `jumlah_transaksi_per_pelanggan` AS  
. SELECT  
.     DATE_FORMAT(`tanggal`, '%Y-%m-%d') AS 'tanggal',  
.     `id_customer`,  
.     `nama_barang`,  
.     `cabang_sales`,  
.     COUNT(*) AS 'jumlah_transaksi'  
. FROM  
.     `data_mart`  
30 WHERE  
.     DATE_FORMAT(`tanggal`, '%Y-%m-%d') BETWEEN '2022-04-01' AND '2022-04-31'  
. GROUP BY  
.     `tanggal`, `id_customer`;  
.
```

Table Aggregate “<< Jumlah_Transaksi_per_pelanggan >>”



The screenshot shows a database management interface with a left sidebar containing a tree view of database objects. The main area displays a table with 6 columns: tanggal, id_customer, nama_barang, cabang_sales, and jumlah_transaksi. The table contains 20 rows of data, representing transactions for various customers and products. The table is titled 'Jumlah_Transaksi_per_pelanggan' in the sidebar.

tanggal	id_customer	nama_barang	cabang_sales	jumlah_transaksi
2022-04-01	CUST55602	AMPICILLIN	Bekasi	1
2022-04-02	CUST55603	TRAMADOL KAP	Padang	1
2022-04-03	CUST55604	KLORPROMAZIN	Lampung	1
2022-04-04	CUST55605	KETOCONAZOLE	Aceh	1
2022-04-05	CUST55606	ERGOTAMINE CC	Kuningan	1
2022-04-06	CUST55607	TETRACYCLINE I	Jakarta	1
2022-04-07	CUST55608	AMBROXOL HC	Bandung	1
2022-04-08	CUST55609	PARACETAMOL	Tangerang	1
2022-04-09	CUST55610	ACYCLOVIR DUS	Bekasi	1
2022-04-10	CUST55611	ALERGINE TABL	Padang	1
2022-04-11	CUST55612	AMPICILLIN	Aceh	1
2022-04-12	CUST55613	TRAMADOL KAP	Kuningan	1
2022-04-13	CUST55614	KLORPROMAZIN	Jakarta	1
2022-04-14	CUST55615	KETOCONAZOLE	Bandung	1
2022-04-15	CUST55616	ACYCLOVIR DUS	Tangerang	1
2022-04-16	CUST55617	ALERGINE TABL	Bekasi	1
2022-04-16	CUST55638	AMBROXOL HC	Bandung	1
2022-04-17	CUST55618	AMPICILLIN	Padang	1
2022-04-17	CUST55639	PARACETAMOL	Tangerang	1
2022-04-18	CUST55619	TRAMADOL KAP	Lampung	1
2022-04-18	CUST55640	ACYCLOVIR DUS	Bekasi	1
2022-04-18	CUST55658	AMBROXOL HC	Kuningan	1
2022-04-18	CUST55661	ALERGINE TABL	Tangerang	1
2022-04-19	CUST55620	ACYCLOVIR DUS	Aceh	1
2022-04-19	CUST55641	ALERGINE TABL	Padang	1
2022-04-19	CUST55659	PARACETAMOL	Jakarta	1
2022-04-19	CUST55662	AMPICILLIN	Bekasi	1
2022-04-20	CUST55621	ALERGINE TABL	Kuningan	1
2022-04-20	CUST55642	AMPICILLIN	Aceh	1

Table Aggregate “<< Jumlah_Transaksi_per_pelanggan >>”

column	data type	description	transformation
Tanggal	Date	Tanggal transaksi dilakukan	DATE FORMAT (tanggal, %y-%m-%d)
Id_customer	String	Id customer, key join ke tabel pelanggan	-
Nama_barang	String	Nama barang	-
Cabang_sales	String	Cabang sales	-
Jumlah_transaksi	Numeric	Jumlah transaksi pelanggan	COUNT(*)

Table Aggregate “<< Pendapatan_per_bulan >>”

```

-- Pendapatan per bulan
CREATE TABLE `pendapatan_per_bulan` AS
SELECT
40     DATE_FORMAT(`tanggal`, '%Y-%m-%d') AS 'tanggal',
        `id_barang`,
        `nama_barang`,
        `unit`,
        `harga`,
        `mata_uang`,
        `jumlah_barang`,
        SUM(`harga` * `jumlah_barang`) AS 'total_pendapatan'
FROM
        `data_mart`
50 WHERE
        DATE_FORMAT(`tanggal`, '%Y-%m-%d') BETWEEN '2022-06-01' AND '2022-06-31'
GROUP BY
        `tanggal`, `nama_barang`;

```

Table Aggregate “<< Pendapatan_per_bulan >>”

Quick access		0	100	Filter					
		tanggal	id_barang	nama_barang	unit	harga	mata_uang	jumlah_barang	total_pendapatan
localhost		2022-06-01	BRG0002	ALERGINE TABLET	DUS	3992 IDR		9	35928
> information_schema		2022-06-02	BRG0003	AMPICILLIN	DUS	5781 IDR		1	5781
> kimia_farma		2022-06-03	BRG0004	TRAMADOL KAPSUL	DUS	6940 IDR		23	159620
> barang		2022-06-04	BRG0001	ACYCLOVIR DUS DUS		1170 IDR		4	4680
> data_mart		2022-06-05	BRG0002	ALERGINE TABLET	DUS	2338 IDR		2	4676
> jumlah_transaksi_per_pelanggan		2022-06-06	BRG0003	AMPICILLIN	DUS	10691 IDR		17	181747
> pelanggan		2022-06-07	BRG0004	TRAMADOL KAPSUL	DUS	8701 IDR		11	95711
> pendapatan_per_bulan		2022-06-08	BRG0005	KLORPROMAZIN DUS		5648 IDR		12	67776
> penjualan		2022-06-09	BRG0006	KETOCONAZOLE DUS		2819 IDR		12	33828
> total_penjualan_per_produk		2022-06-10	BRG0007	ERGOTAMINE CLOMIDAT	DUS	4592 IDR		25	114800
> mysql		2022-06-11	BRG0008	TETRACYCLINE DUS		3992 IDR		14	55888
> performance_schema		2022-06-12	BRG0009	AMBROXOL HC DUS		5781 IDR		34	196554
> phpmyadmin		2022-06-13	BRG0010	PARACETAMOL DUS		6940 IDR		67	464980
> test		2022-06-14	BRG0001	ACYCLOVIR DUS DUS		1170 IDR		15	17550
Proses - Proses		2022-06-15	BRG0002	ALERGINE TABLET	DUS	2338 IDR		12	28056
User		2022-06-16	BRG0002	ALERGINE TABLET	DUS	3992 IDR		23	91816
Variabel - Variabel		2022-06-16	BRG0003	AMPICILLIN	DUS	10691 IDR		25	812516
		2022-06-16	BRG0008	TETRACYCLINE DUS		3992 IDR		12	211576

Table Aggregate “<< Pendapatan_per_bulan >>”

column	data type	description	transformation
Tanggal	Date	Tanggal transaksi dilakukan	DATE FORMAT (tanggal, %y-%m-%d)
Id_barang	String	Id barang	-
Nama_barang	String	Nama barang	-
Unit	String	Unit barang	-
Harga	Numeric	Harga barang	-
Mata_uang	String	Mata uang, default = IDR	-
Jumlah_barang	Numeric	Jumlah barang	-
Total_pendapatan	Numeric	Total pendapatan per bulan	SUM (harga * jumlah_barang)

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

Jawaban :

Link visualisasi (ex link Google Data Studio) :

<https://lookerstudio.google.com/reporting/fd197f5b-c581-4f41-a34c-8aa401ab6476>

Data Mart - Kimia Farma

Produk

Pelanggan

Penjualan

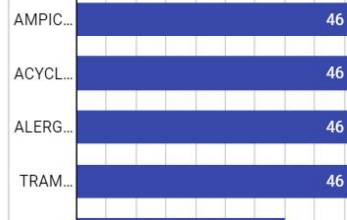
Data Barang

Nama Produk

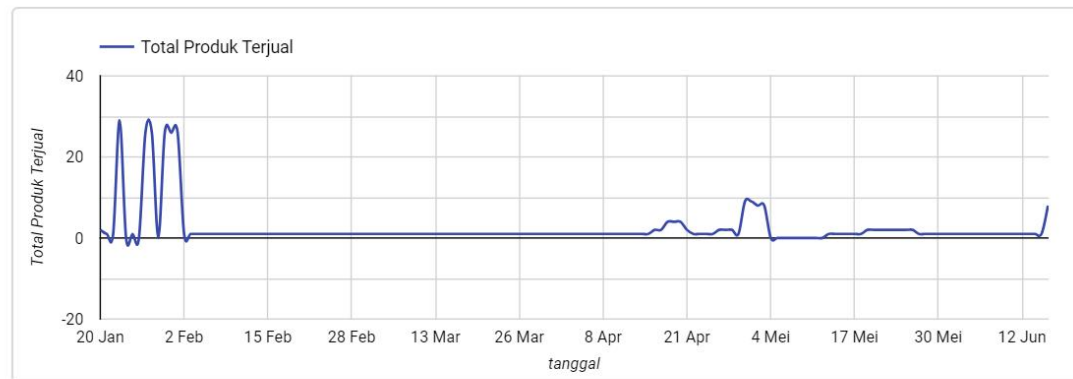
Pilih rentang tanggal

Total Produk Terjual

9.176



Trend penjualan produk



Data Mart - Kimia Farma

Produk

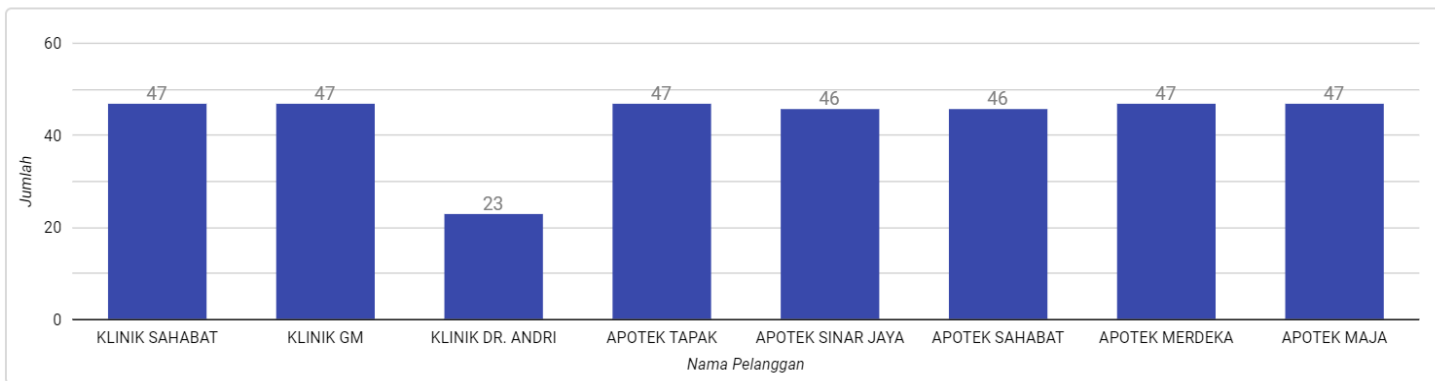
Pelanggan

Penjualan

Data Pelanggan

Pilih rentang tanggal

Top Pelanggan



Top group

Top pelanggan berdasarkan wilayah

Data Mart - Kimia Farma

Produk

Pelanggan

Penjualan

Data Penjualan

cabang_sales

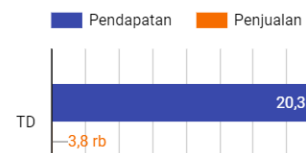
Nama Produk

Pilih rentang tanggal

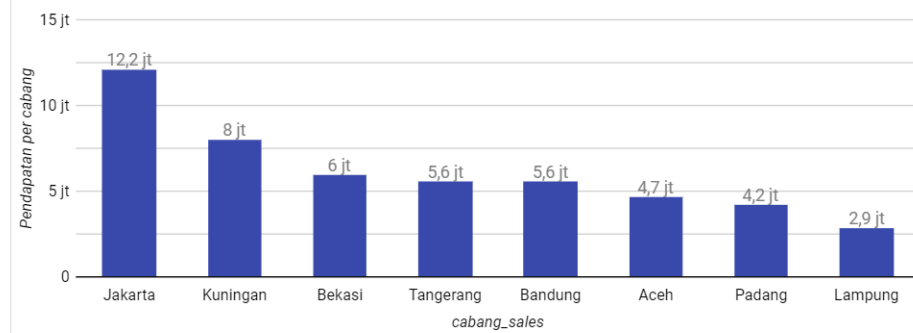
Total Pendapatan

49.217.151

Top Distributor



Pendapatan per cabang



Soal 6 : Additional Complementary Data

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

Jawaban :

Data latitude longitude untuk setiap lokasi pelanggan

- a. tujuannya untuk melihat persebaran lokasi pelanggan
- b. dari data tersebut, kita bisa melihat potential lokasi yang bisa kita approach untuk menjadi pelanggan.
- c. selain itu, kita juga bisa memprediksi demand lokasi 1A ke lokasi sekitar dari kebutuhan obat di lokasi tertentu pada waktu tertentu. misalkan sedang musim pengujian di daerah A sehingga kebutuhan obat di sekitar daerah tersebut terkait dengan obat flu atau demam. dan kita bisa mengalokasikan stok obat pada distributor / cabang tertentu terhadap demand prediction tersebut

Data cuaca

- a. untuk melihat korelasi antara kebutuhan obat dengan cuaca di daerah tersebut