**MY SQL QUERIES**

**COFFEE SHOP SALES PROJECT**

STEP FOR MY SQL

* Data Walkthrough (Panduan Data)
* Raw data file preparation (Persiapan Data Mentah)
* Creating Database (Membuat Database)
* Importing File (Mengimport File)
* Cleaning Imported File (Membersihkan File yang telah diimpor)
* Changing Data Types (Mengubah Tipe Data)
* Firing SQL Queries for Business Requirements (Menjalankan Query SQL untuk Kebutuhan Bisnis)
* Storing Results (Menyimpan hasil)
* Preparing SQL Documents (Menyiapkan dokumen SQL)

FUNGSI-FUNGSI YANG AKAN DIPELAJARI

|  |  |
| --- | --- |
| * STR\_TO\_DATE | : Mengubah tipe string menjadi format tanggal |
| * ROUND | : Membulatkan angka ke jumlah desimal tertentu |
| * SUM | : Menjumlahkan nilai dalam suatu kolom |
| * COUNT | : Menghitung jumlah baris dalam suatu kolom |
| * AVG | : Menghitung rata-rata nilai dalam suatu kolom |
| * LAG | : Mengambil nilai dari baris sebelumnya dalam kumpulan data |
| * MONTH | : Mengambil nilai bulan dari suatu tanggal |
| * DAY | : Mengambil nilai hari dari suatu tanggal |
| * DAYOFWEEK | : Mengambil informasi hari dalam format numerik (1 = minggu, 2 = senin, dll) |
| * SELECT | : Digunakan untuk mengambil data dari tabel |
| * ALIAS / AS | : Memberikan nama lain pada kolom atau tabel dalam query |
| * MAX/MIN | : Mengambil nilai maksimum atau minimum dalam suatu kolom |
| * HOUR | : Mengambil informasi jam dari suatu tanggal/waktu |
| * ALTER TABLE | : Mengubah struktur tabel (menambah, mengubah, atau menghapus kolom) |
| * UPDATE TABLE | : Mengubah nilai dari suatu tabel |
| * CHANGE COLUMN | : Mengubah tipe data atau nama kolom dalam tabel |
| * WHERE | : Digunakan untuk memfilter data berdasarkan kondisi tertentu |
| * GROUP BY | : Mengelompokkan data berdasarkan satu atau lebih kolom |
| * CASE | : Menggunakan kondisi if-else dalam SQL |
| * ORDER BY | : Mengurutkan data berdasarkan kolom tertentu |
| * LIMIT | : Membatasi jumlah baris yang ditampilkan dalam hasil query |
| * WIDOW FUNCTION | : Digunakan untuk melakukan operasi analitik pada kumpulan data (misalnya LAG, LEAD) |
| * JOINS | : Menggabungkan data dari dua atau lebih tabel berdasarkan hubungan tertentu |
| * SUBQUERIES | : Query yang terdapat dalam query lain (digunakan untuk pengambilan data yang lebih kompleks) |

1. **Pembuatan dan Pemanggilan Database**
   1. **Membuat Database**

CREATE DATABASE coffee\_shop\_sales

Membuat database baru Bernama **coffee\_shop\_sales** untuk menyimpan data transaksi penjualan di coffee shop

* 1. **Memanggil Database**

SELECT \* FROM coffee\_shop\_sales;

Menampilkan seluruh data dalam tabel **coffee\_shop\_sales** setelah database dibuat dan data telah dimasukkan.

1. **Data Cleaning**
   1. **Melihat Struktur Data**

DESCRIBE coffee\_shop\_sales;

Menampilkan informasi struktur tabel **coffee\_shop\_sales,** termasuk nama kolom, tipe data, dan atribut lainnya.

* 1. **Menonaktifkan Mode Keamanan SQL**

SET SQL\_SAFE\_UPDATES = 0;

Mengizinkan perubahan data pada tabel tanpa Batasan keamanan ketat.

* 1. **Mengubah Tipe Data *transaction\_date* dari Text ke Date**

UPDATE coffee\_shop\_sales

SET transaction\_date = STR\_TO\_DATE(transaction\_date, '%d/%m/%Y');

* *UPDATE coffee\_shop\_sales* Perintah ini menunjukkan bahwa kita ingin memperbarui (update) data dalam tabel **coffee\_shop\_sales**
* *SET transaction\_date = …* Kolom **transaction\_date** akan diperbaharui nilainya
* *STR\_TO\_DATE(transaction\_date, '%d/%m/%Y')*  mengonversi string (teks) menjadi format **Date** dari kolom **transaction\_date** menjadi format %d%m%Y

ALTER TABLE coffee\_shop\_sales

MODIFY COLUMN transaction\_date DATE;

* *ALTER TABLE coffee\_shop\_sales* Mengubah struktur tabel **coffee\_shop\_sales**
* *MODIFY COLUMN transaction\_date DATE* Digunakan untuk mengubah definisi kolom yang sudah ada, mengubah kolom **transaction\_date** ke tipe **Date**
  1. **Mengubah Tipe Data *transaction\_time* dari Text ke Time**

UPDATE coffee\_shop\_sales

SET transaction\_time = STR\_TO\_DATE(transaction\_time, '%H:%i:%s');

* Mengonversi data **transaction\_time** dari format teks ke format waktu (TIME)

ALTER TABLE coffee\_shop\_sales

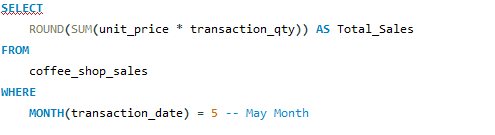
MODIFY COLUMN transaction\_time TIME;

* Mengubah tipe data dari kolom **transaction\_time** menjadi TIME secara permanen dalam tabel
  1. **Memeriksa Perubahan Struktur Data**

DESCRIBE coffee\_shop\_sales;

Menampilkan Kembali struktur tabel setelah perubahan tipe data dilakukan

1. **Analisis KPI Penjualan**
   1. **Analisis Total Penjualan** 
      1. **Menghitung total penjualan untuk setiap bulan**

****

SELECT

Memilih kolom yang akan ditampilkan

ROUND(SUM(unit\_price \* transaction\_qty)) AS Total\_Sales

* *SUM(unit\_price\*transaction\_qty))*  menghitung total penjualan dengan mengalikan harga per unit dengan jumlah unit yang terjual
* *ROUND(…)* Membulatkan hasil total penjualan ke bilangan bulat terdekat
* *AS Total\_Sales* memberi alias **Total\_Sales** pada hasil query agar lebih mudah dibaca

FROM

coffee\_shop\_sales

Data diambil dari tabel **coffee\_shop\_sales**, yang menyimpan informasi transaksi

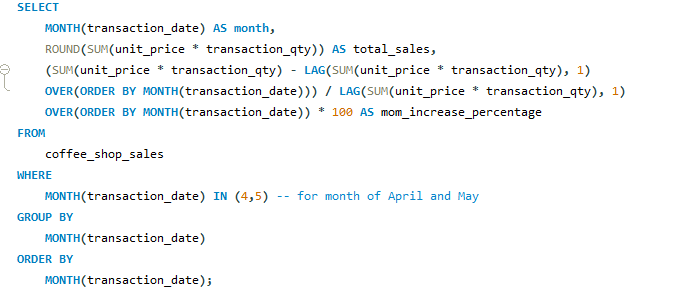
WHERE Kondisi penyaringan kata

MONTH(transaction\_date) = 5; -- May month

* *MONTH(transaction\_date) = 5* Mengambil angka bulan dari kolom **transaction\_date** di Bulan Mei (5). Memfilter hanya transaksi yang terjadi di bulan Mei



* + 1. **Menentukan Peningkatan atau Penurunan Penjualan dari Bulan ke Bulan**

****

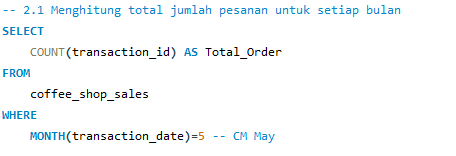
* LAG(…) untuk mengambil nilai total penjualan bulan sebelumnya

**Total Sales bulan Sebelumnya**

LAG(SUM(unit\_price \* transaction\_qty), 1) OVER(ORDER BY MONTH(transaction\_date))

* Fungsi LAG(…) mengambil nilai total bulan sebelumnya
* *OVER(ORDER BY MONTH(transaction\_date))* memastikan data diurutkan berdasarkan bulan
* *FROM coffee\_shop\_sales* Mengambil data dari tabel coffee\_shop\_sales
* *WHERE MONTH(transaction\_date) IN (4,5)* Hanya mengambil data untuk bulan April (4) dan Mei (5)
* *GROUP BY MONTH(transaction\_date)* Data dijumlahkan berdasarkan bulan (MONTH(transaction\_date)) agar total penjualan dihitung untuk masing-masing bulan
* *ORDER BY MONTH(transaction\_date))* Mengurutkan hasil berdasarkan urutan bulan (April Mei)

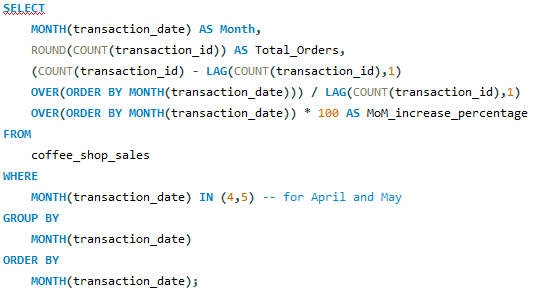


* 1. **Analisis Total Pesanan**
     1. **Menghitung total jumlah pesanan untuk setiap bulan**

Memilih Kolom dengan menghitung jumlah baris dari kolom **transaction\_id** sebagai Total\_Order dari tabel **coffee\_shop\_sales** dimana bulan yang dipilih dari kolom **transaction\_date** adalah bulan Mei (5).

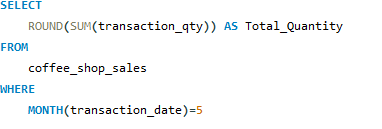


* + 1. **Menentukan peningkatan atau penurunan jumlah pesanan dari bulan ke bulan**

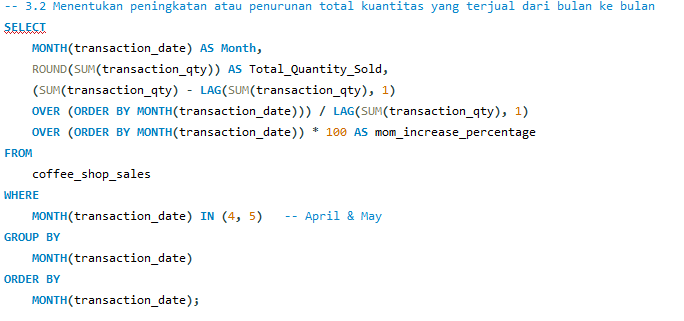
****

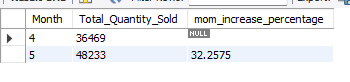


* 1. **Analisis Total Kuantitas Terjual**

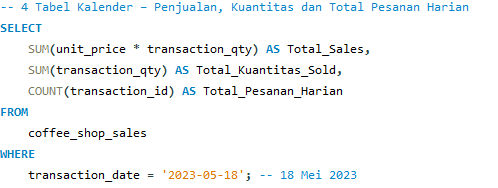








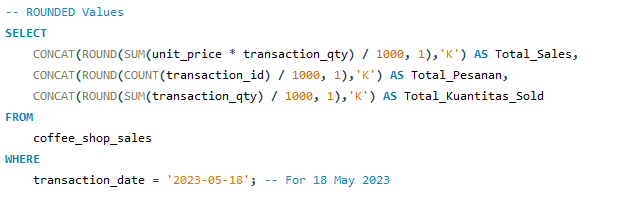
1. **Analisis Grafik Penjualan**
   1. **Tabel Kalender – Penjualan, Kuantitas dan Total Pesanan Harian**

****

**Output**

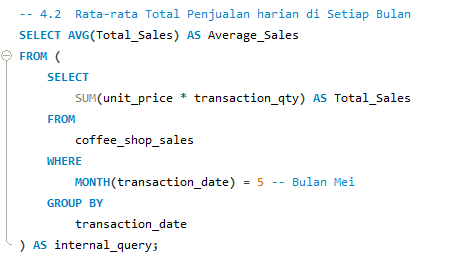


**VERSI ROUND (Pembulatan)**

****

**Output**



* 1. **Rata-rata Total Penjualan haria di Setiap Bulan**

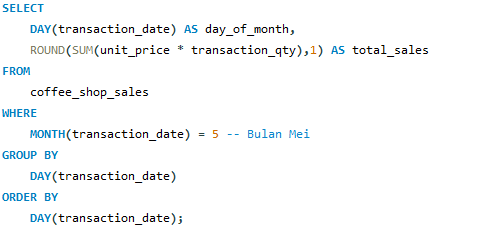
**QUERY UTAMA**

* *AVG(Total\_Sales)* Menghitung rata-rata dari **total\_sales** yang digitung di subquery
* *AS average\_sales* memberikan alias untuk hasil query agar lebih mudah dibaca

**SUBQUERY (internal\_query)**

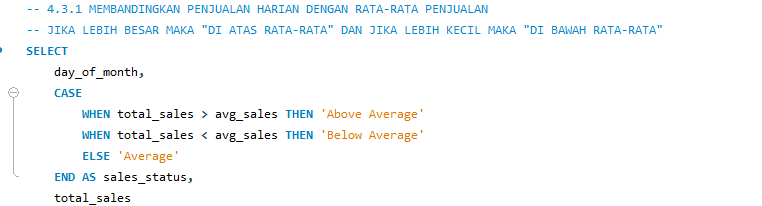
* Menghitung total penjualan per hari di bulan Mei
* Mengelompokkan data berdasarkan **transaction\_date** agar setiap hari memiliki satu total penjualan
* Menghasilkan dataset sementara (**internal\_query)** yang berisi total penjualan harian, yang kemudian digunakan dalam query utama untuk menghitung rata-rata

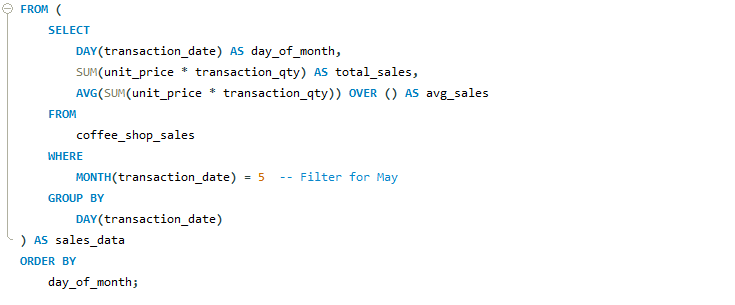


* 1. **Penjualan Harian untuk Bulan yang Dipilih (Bulan Mei)**



**Membandingkan Penjualan Harian dengan Rata-Rata Penjualan – Jika Lebih Besar Maka “Above Average” dan Jika Lebih Kecil maka “Below Average”**

****

****

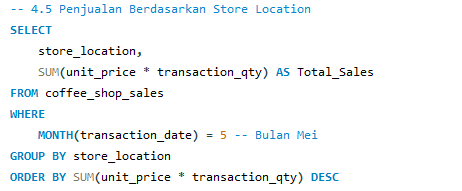
**Output**



* 1. **Penjualan Berdasarkan Weekday / Weekend**
* **CASE** : Digunakan karena akan menggunakan kondisional if-else
* **DAYOFWEEK(transaction\_date)** : Mengambil kode hari dalam seminggu berdasarkan **transaction\_date**, dengan aturan apabila (1,7) (1=minggu, 7=sabtu) maka diberi keterangan ‘Weekends’, kalua tidak ‘Weekdays’**. Hasilnya** disimpan sebagai **day\_type**
* **ROUND(SUM(unit\_price\*transaction\_qty),2)** : Menghitung total penjualan dengan mengalikan unit price dengan transaction qty. Kemudian hasilnya dibulatkan **2 desimal**. Kemudian **Hasilnya** disimpan sebagai **total\_sales**
* **FROM** Dari Tabel **coffee\_shop\_sales**
* **WHERE** (memfilter data) hanya memilih data pada bulan ke-5 (mei)
* **GROUP BY** (Mengelompokkan Data Berdasarkan Jenis Hari)

**Output**

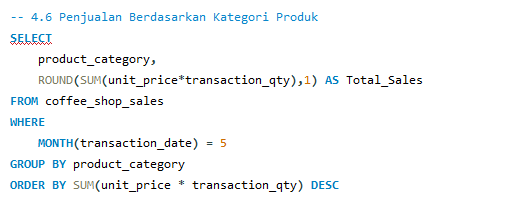
* 1. **Penjualan Berdasarkan Store Location**

****

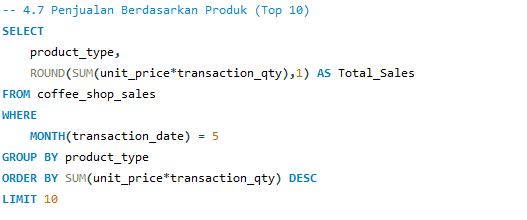
* **Store\_location** menampilkan lokasi toko
* **WHERE** (Memfilter Data) memfilter hanya transaksi yang terjadi di bulan Mei
* **GROUP BY** Mengelompokkan data berdasarkan **store\_location**
* **ORDER BY** (mengurutkan Data) Mengurutkan hasil berdasarkan total penjualan dari **Tertinggi ke Terendah (DESC : descending)**

**OUTPUT**

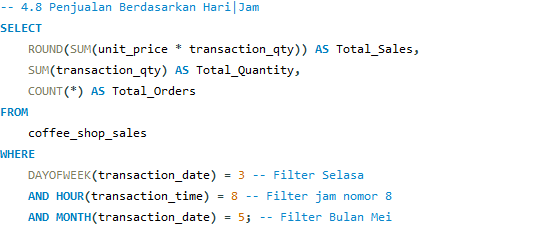
* 1. **Penjualan Berdasarkan Kategori Produk**

****

**Output**

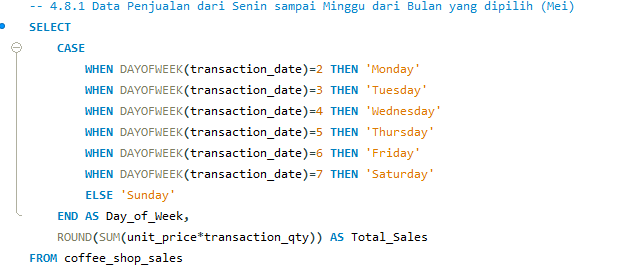
* 1. **Penjualan Berdasarkan Produk (Top 10)**
* **LIMIT 10** : Untuk mengambil hanya 10 data teratas saja

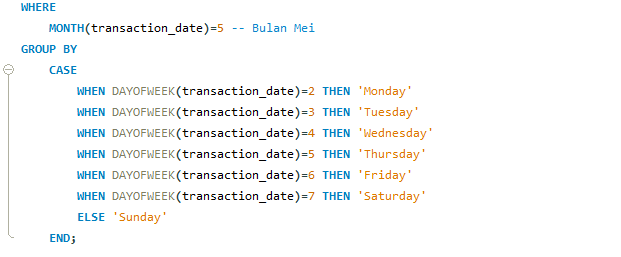
**OUTPUT**

* 1. **Penjualan Berdasarkan Hari | Jam**

**OUTPUT**

**Untuk Mendapatkan Data Penjualan Dari Senin sampai Minggu untuk Setiap Bulan (Bulan Mei)**

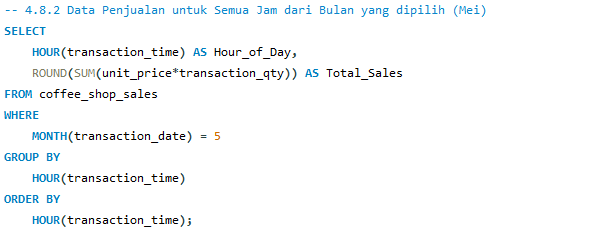


****

**OUTPUT**



**Untuk Mendapatkan Data Penjualan untuk Semua Jam Pada Bulan yang Dipilih (Mei)**

****

**OUTPUT**

