

Nama :maulana ihsan

Nim : 23241049

```
-- fungsi agregasi
-- ambil untuk mengolah beberapa baris/row data untuk menghasilkan nilai baru
-- praktek 7
-- hitung atau jumlahkan seluruh nilai dari baris dari qty pada tabel penjualan
SELECT qty FROM tr_penjualan;
SELECT SUM(qty) FROM tr_penjualan;
```

- Menghitung total jumlah barang yang terjual berdasarkan kolom qty dari semua baris pada tabel tr_penjualan.
- Cocok untuk melihat total keseluruhan penjualan.

```
-- praktek 8
-- hitung seluruh baris pada tabel penjualan
SELECT * FROM tr_penjualan;
SELECT COUNT(*) FROM tr_penjualan;
```

- Menghitung jumlah total baris data dalam tabel tr_penjualan.
- Berguna untuk mengetahui berapa transaksi yang telah terjadi.

```
-- praktek 9
-- mengambil 2 fungsi agregasi dalam satu query sql
SELECT SUM(qty), COUNT(*) FROM tr_penjualan;
```

Menggabungkan dua fungsi agregasi dalam satu query:

- SUM(qty) menghitung total barang yang terjual.
- COUNT(*) menghitung jumlah transaksi.

```
-- praktek 10
-- hitung rata rata penjualan, penjualan tertinggi per transaksi, dan penjualan terendah per transaksi
• SELECT AVG(qty), MAX(qty), MIN(qty)
FROM tr_penjualan;
```

- AVG(qty) menghitung rata-rata barang yang terjual per transaksi.
- MAX(qty) mencari jumlah penjualan terbanyak dalam satu transaksi.
- MIN(qty) mencari penjualan terkecil dalam satu transaksi.

```
-- praktek 11
-- mengkombinasikan fungsi agregasi dengan kolom tertentu (nama_produk, kolom tanggal transaksi)
SELECT tgl_transaksi, nama_produk, MAX(qty) FROM tr_penjualan;
```

- Kombinasi kolom biasa (tgl_transaksi, nama_produk) dengan agregasi (MAX(qty)) harus menggunakan GROUP BY, jika tidak akan error.
- Tujuannya untuk mencari jumlah maksimal per kombinasi tanggal dan nama produk.

```
-- COUNT(DISTINCT)
-- digunakan untuk menghitung nilai yang unik saja
-- menghitung nilai unik yang ada pada nama produk di tabel penjualan
-- tanpa distinct
-- praktek 12
SELECT COUNT(nama_produk) FROM tr_penjualan;
```

```
-- praktek 13
-- dengan distinct
SELECT COUNT(distinct nama_produk) FROM tr_penjualan;
SELECT COUNT(nama_produk), COUNT(distinct nama_produk) FROM tr_penjualan;
```

- COUNT(nama_produk) menghitung **semua baris** yang memiliki nilai pada kolom nama_produk.
- COUNT(DISTINCT nama_produk) menghitung **jumlah produk yang berbeda/unik**.
- Sangat bermanfaat untuk mengetahui keragaman produk.

```
-- GROUP BY
-- mengelompokkan isi data dari satu atau beberapa kolom, biasanya digabung dengan fungsi agregasi
-- GROUP BY 1 KOLON
-- praktek 14
SELECT nama_produk FROM tr_penjualan GROUP BY nama_produk;
SELECT nama_produk FROM tr_penjualan;
```

- Menampilkan **daftar produk unik** dalam tabel.

```
-- praktek 15
-- mengelompokkan nilai nama produk dan qty pada tabel penjualan
SELECT nama_produk, qty from tr_penjualan
GROUP BY nama_produk, qty;
```

- Mengelompokkan berdasarkan kombinasi nama_produk dan qty.

```
-- praktek 16
-- ambil qty dari nama produk berdasarkan hasil pengelompokan nama produk
• SELECT nama_produk, SUM(qty) FROM tr_penjualan GROUP BY nama_produk;
```

- Mengelompokkan penjualan berdasarkan produk dan menghitung total barang yang terjual untuk tiap produk.

```
-- praktek 17
-- ambil qty dari nama produk berdasarkan hasil pengelompokan nama produk urut dari besar terkecil
SELECT nama_produk, SUM(qty) FROM tr_penjualan GROUP BY nama_produk ORDER BY SUM(qty) DESC;
```

- Mengurutkan hasil berdasarkan total kuantitas dari yang **terbanyak ke yang paling sedikit**.

```
-- HAVING
-- digunakan sebagai filtering untuk GROUP BY dan fungsi agregasi
-- praktek 18
-- ambil jumlah qty
select nama_produk, SUM(qty)
FROM tr_penjualan
GROUP BY nama_produk
HAVING sum (qty) > 2
ORDER BY SUM(qty) DESC;
```

- HAVING digunakan seperti WHERE, tapi khusus untuk hasil **setelah pengelompokan dan agregasi**.
- WHERE tidak bisa digunakan pada hasil agregat, maka digunakan HAVING.

```
-- Tugas mandiri
-- melihat daftar nama produk yang total jumlah produk terjual per kelompok nama produk adalah di atas nilai 4
SELECT
    nama_produk,
    kelompok_produk,
    SUM(jumlah_terjual) AS total_terjual
FROM
    penjualan
GROUP BY
    nama_produk, kelompok_produk
HAVING
    SUM(jumlah_terjual) > 4;
```

- Mengambil produk dengan penjualan lebih dari 4 unit berdasarkan pengelompokan produk dan kelompok produk.

```
-- melihat daftar nama produk yang total jumlah produk terjual perkelompok nama produk sama dengan 9
SELECT
    nama_produk,
    kelompok_produk,
    SUM(jumlah_terjual) AS total_terjual
FROM
    penjualan
GROUP BY
    nama_produk, kelompok_produk
HAVING
    SUM(jumlah_terjual) = 9;
```

- Menampilkan produk yang total jumlah terjualnya tepat **9 unit**.

```
-- melihat daftar kelompok nama produk dan total nilai penjualan (harga dikalikan jumlah dikurangi diskon) dengan dan urutan berdasarkan nil
SELECT
    kelompok_produk,
    nama_produk,
    SUM((harga * jumlah_terjual) - diskon) AS total_penjualan
FROM
    penjualan
GROUP BY
    kelompok_produk, nama_produk
ORDER BY
    total_penjualan DESC;
```

- Menghitung **total pendapatan** berdasarkan rumus:
 - o **Harga × jumlah dikurangi diskon**

- Lalu hasilnya diurutkan dari yang **tertinggi ke terendah**.