**Day 11: SQL Session 1**

Github | [**Langsung ke tugas**](#tugas)

**1. Membuat Database dan Tabel**

Setelah membuat database, mari kita buat tabel dengan SQL dibawah:

|  |
| --- |
| -- Create Table  CREATE TABLE IF NOT EXISTS customer(      customer\_id INT(11) NOT NULL AUTO\_INCREMENT,      PRIMARY KEY (customer\_id),      customer\_name VARCHAR(20) NOT NULL,      customer\_city VARCHAR(20) NOT NULL  );  CREATE TABLE IF NOT EXISTS salesman(      salesman\_id INT(11) NOT NULL AUTO\_INCREMENT,      PRIMARY KEY (salesman\_id),      salesman\_name VARCHAR(20) NOT NULL,      salesman\_city VARCHAR(20) NOT NULL,      salesman\_commission INT(11)  );  CREATE TABLE IF NOT EXISTS orders(      orders\_id INT(11) NOT NULL AUTO\_INCREMENT,      PRIMARY KEY (orders\_id),      orders\_date DATE NOT NULL,      orders\_amount INT(11) NOT NULL,      customer\_id INT(11) NOT NULL,      FOREIGN KEY (customer\_id) REFERENCES customer(customer\_id),      salesman\_id INT(11) NOT NULL,      FOREIGN KEY (salesman\_id) REFERENCES salesman(salesman\_id)  ); |

Berikut hasil dari kueri SQL diatas:

|  |
| --- |
|  |

**2. Memasukkan data**

Pertama, masukkan data untuk customer dan salesman terlebih dahulu:

|  |
| --- |
| -- Data Customer  INSERT INTO customer VALUES(      NULL,      'Elskan Wilde',      'JAKARTA'  );  INSERT INTO customer VALUES(      NULL,      'Aprilya Yusri',      'JAKARTA'  );  INSERT INTO customer VALUES(      NULL,      'Rio Novaldi',      'BANDUNG'  );  INSERT INTO customer VALUES(      NULL,      'Ezar Nolan',      'BANDUNG'  );  INSERT INTO customer VALUES(      NULL,      'Jessen Cornelius',      'BANDUNG'  );  -- Data Salesman  INSERT INTO salesman VALUES(      NULL,      'Farhan Harahap',      'JAKARTA',      NULL  );  INSERT INTO salesman VALUES(      NULL,      'Dzaki Marhaen',      'BANDUNG',      NULL  ); |

Kedua, masukkan data untuk orders:

|  |
| --- |
| -- Data Orders  INSERT INTO orders VALUES(      NULL,      '2024-2-23',      80,      1,      1  );  INSERT INTO orders VALUES(      NULL,      '2024-2-24',      125,      1,      1  );  INSERT INTO orders VALUES(      NULL,      '2024-2-24',      40,      4,      2  );  INSERT INTO orders VALUES(      NULL,      '2024-2-25',      65,      3,      2  );  INSERT INTO orders VALUES(      NULL,      '2024-2-25',      75,      4,      2  );  INSERT INTO orders VALUES(      NULL,      '2024-2-25',      25,      3,      2  );  INSERT INTO orders VALUES(      NULL,      '2024-2-25',      50,      1,      1  ); |

Sehingga, data siap digunakan:

|  |
| --- |
|  |

**3. Presentasi Kueri Tugas**

Nomor 1: Tentukanlah pelanggan yang tidak pernah membuat pesanan!

|  |
| --- |
| -- 1. Tentukanlah pelanggan yang tidak pernah membuat pesanan!  SELECT \* FROM customer  WHERE customer\_id NOT IN (      SELECT customer.customer\_id      FROM customer      RIGHT JOIN orders      ON customer.customer\_id = orders.customer\_id  ); |

Nomor 2: Tentukan total banyak pembelian yang dilakukan oleh setiap pelanggan!

|  |
| --- |
| -- 2. Tentukan total banyak pembelian yang dilakukan oleh setiap pelanggan!  SELECT customer.customer\_id, customer.customer\_name, count(\*) AS 'transactions'  FROM orders  LEFT JOIN customer  ON customer.customer\_id = orders.customer\_id  GROUP BY orders.customer\_id; |

Nomor 3: Tentukan nama pelanggan beserta total banyak pesanan yang dilakukan!

|  |
| --- |
| -- 3. Tentukan nama pelanggan beserta total banyak pesanan yang dilakukan!  SELECT customer.customer\_id, customer.customer\_name, SUM(orders.orders\_amount) AS 'total'  FROM orders  LEFT JOIN customer  ON customer.customer\_id = orders.customer\_id  GROUP BY orders.customer\_id; |

Nomor 4: Cari nilai max, min dan rata-rata dari amountnya!

|  |
| --- |
| -- 4. Cari nilai max, min dan rata-rata dari amountnya!  SELECT MIN(total) AS 'min\_total', AVG(total) AS 'avg\_total', MAX(total) AS 'max\_total'  FROM (      SELECT customer.customer\_id, customer.customer\_name, SUM(orders.orders\_amount) AS total      FROM orders      LEFT JOIN customer ON customer.customer\_id = orders.customer\_id      GROUP BY orders.customer\_id  ) AS subquery; |