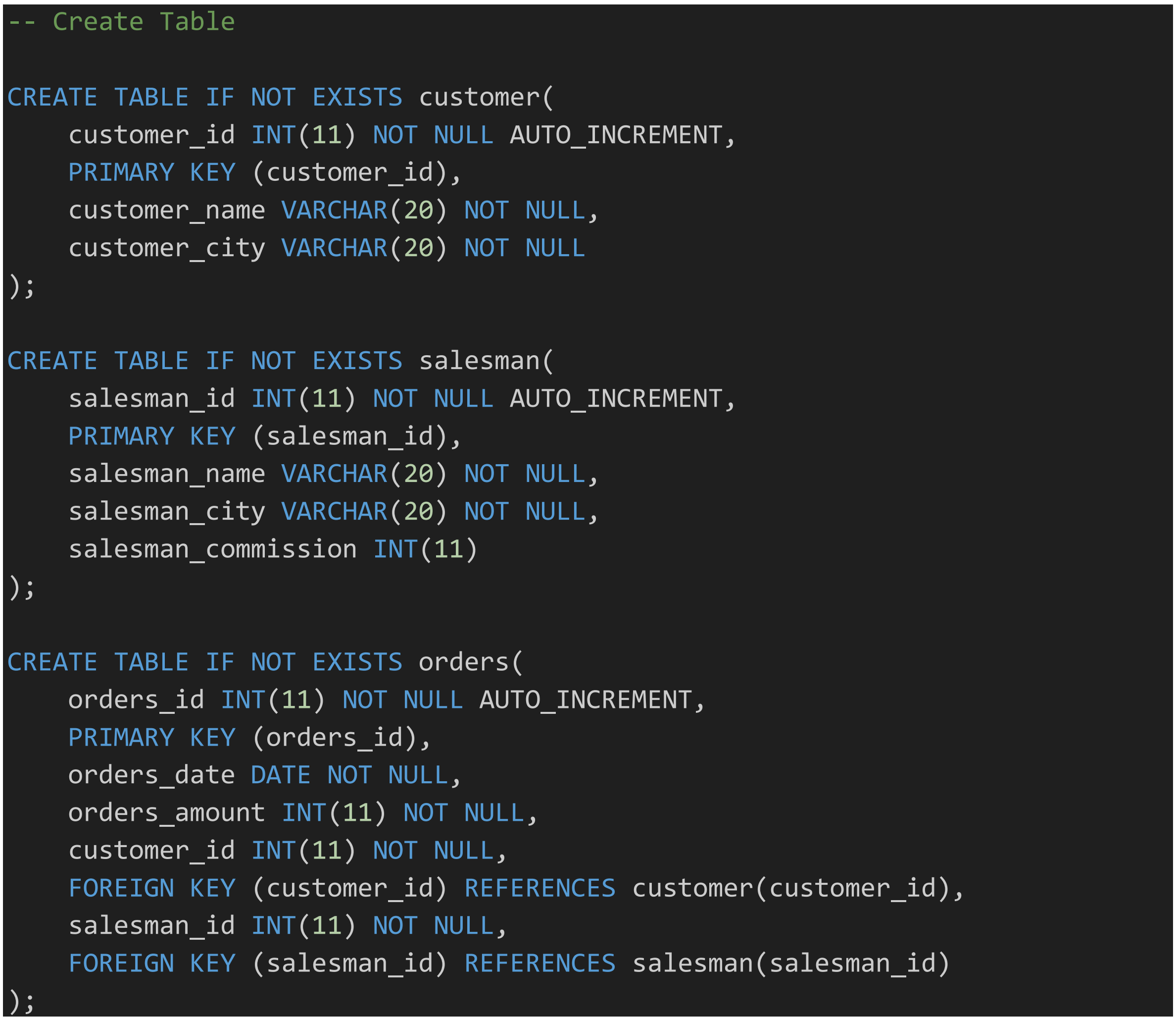
Day 11: SQL Session 1

Langsung ke 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:

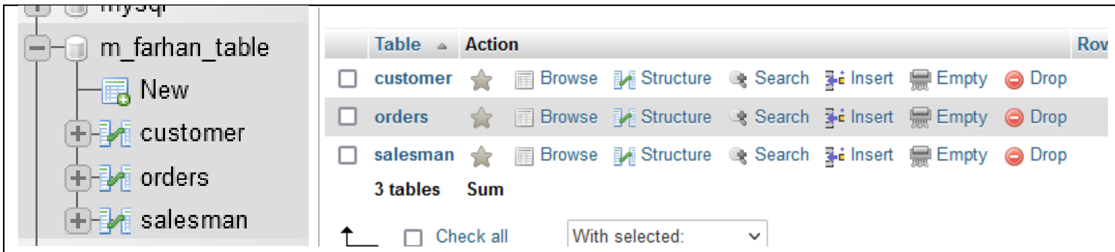


	Table	Action							Rov
□	customer	★	Browse	Structure	Search	Insert	Empty	Drop	
□	orders	★	Browse	Structure	Search	Insert	Empty	Drop	
□	salesman	★	Browse	Structure	Search	Insert	Empty	Drop	
	3 tables	Sum							

□ Check all   With selected:

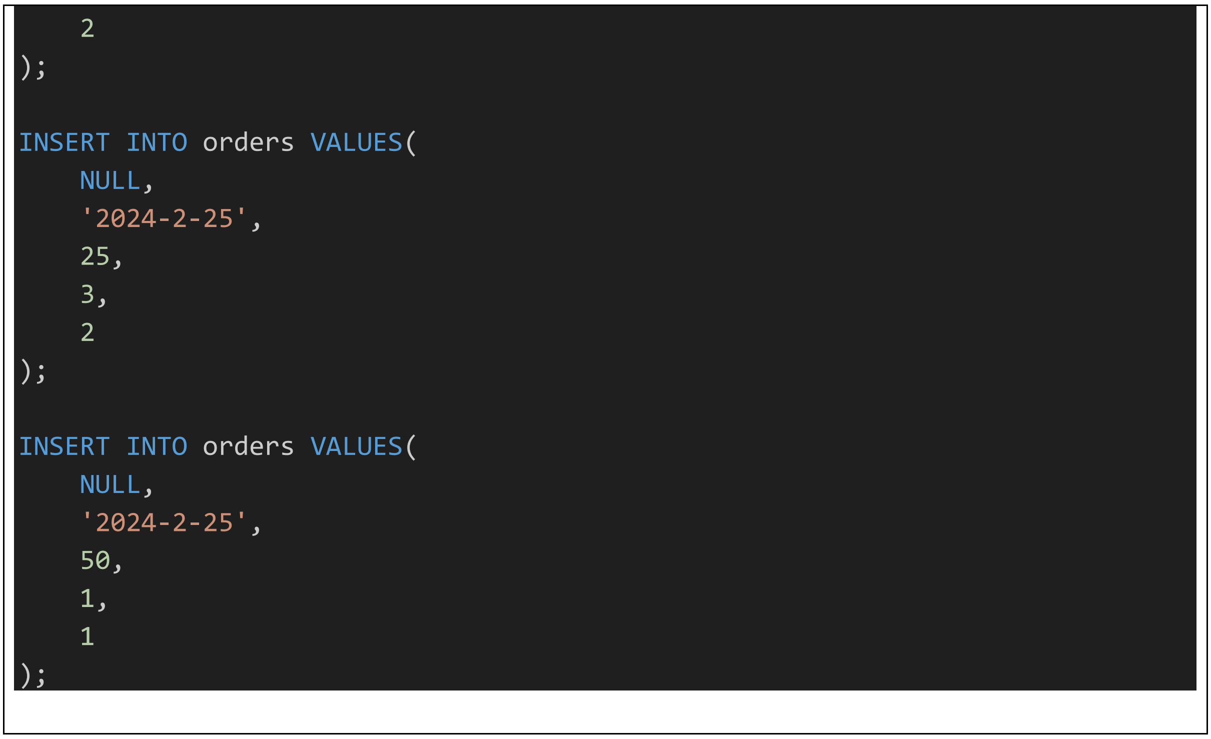
2. Memasukkan data

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
);
```

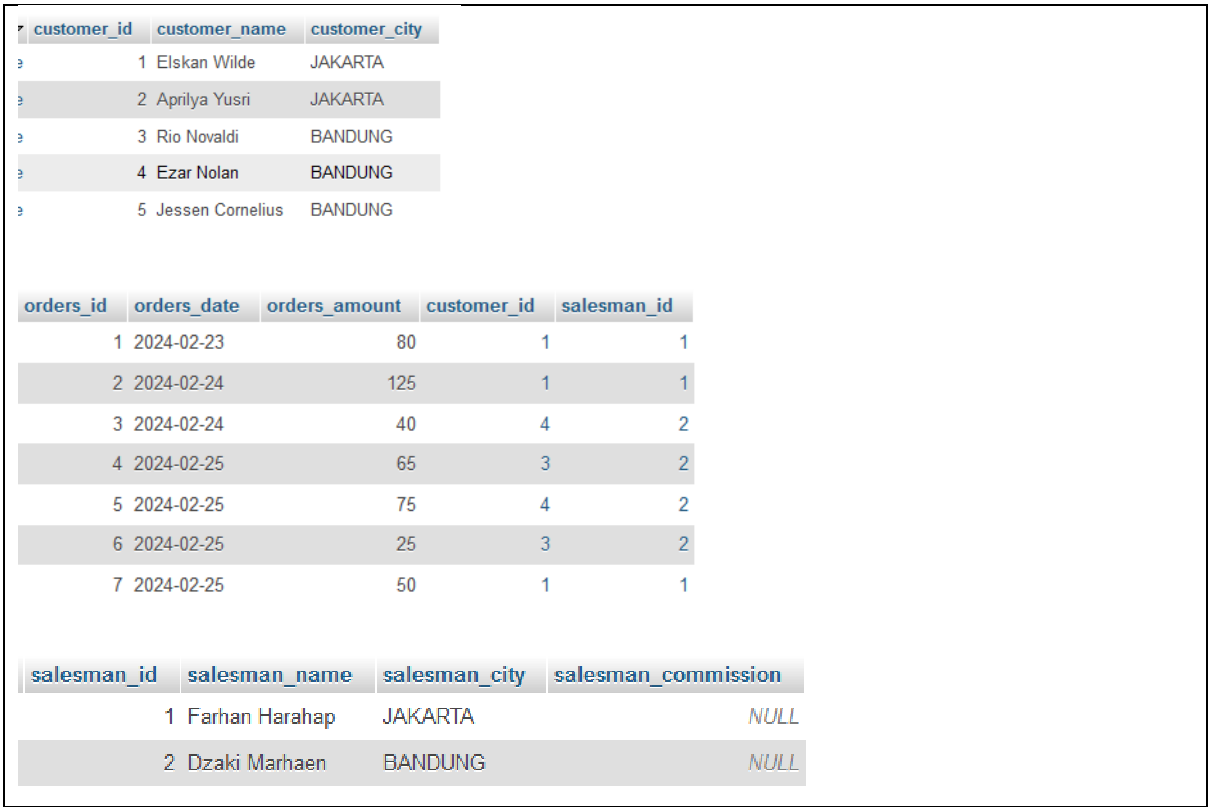
Kedua, masukkan data untuk orders:

```
-- 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:



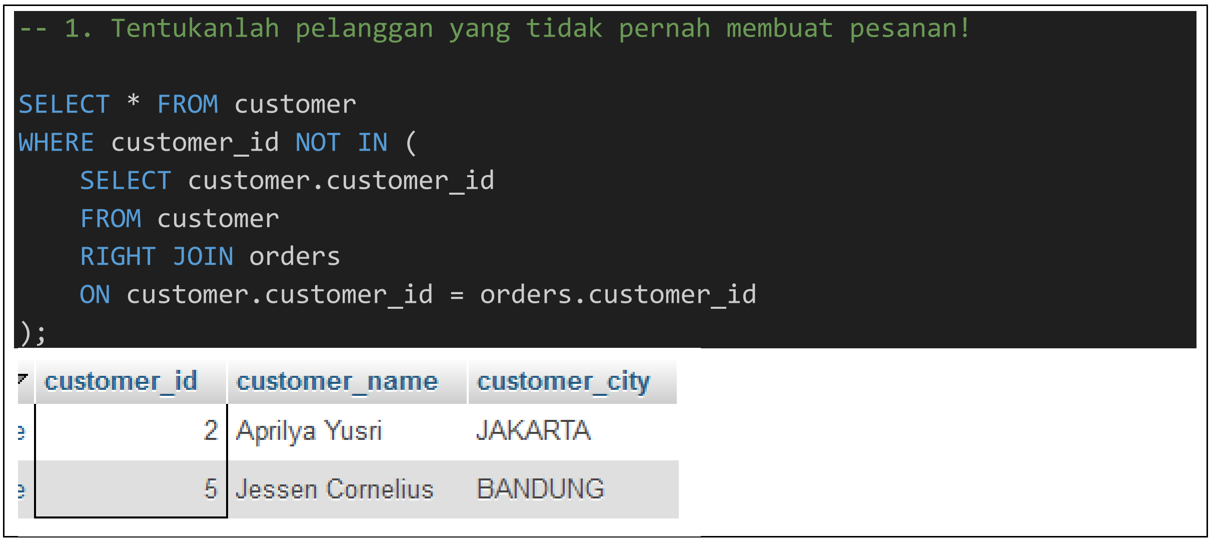
customer_id	customer_name	customer_city
1	Elskan Wilde	JAKARTA
2	Aprilya Yusri	JAKARTA
3	Rio Novaldi	BANDUNG
4	Ezar Nolan	BANDUNG
5	Jessen Cornelius	BANDUNG

orders_id	orders_date	orders_amount	customer_id	salesman_id
1	2024-02-23	80	1	1
2	2024-02-24	125	1	1
3	2024-02-24	40	4	2
4	2024-02-25	65	3	2
5	2024-02-25	75	4	2
6	2024-02-25	25	3	2
7	2024-02-25	50	1	1

salesman_id	salesman_name	salesman_city	salesman_commission
1	Farhan Harahap	JAKARTA	NULL
2	Dzaki Marhaen	BANDUNG	NULL

3. Presentasi Kueri Tugas

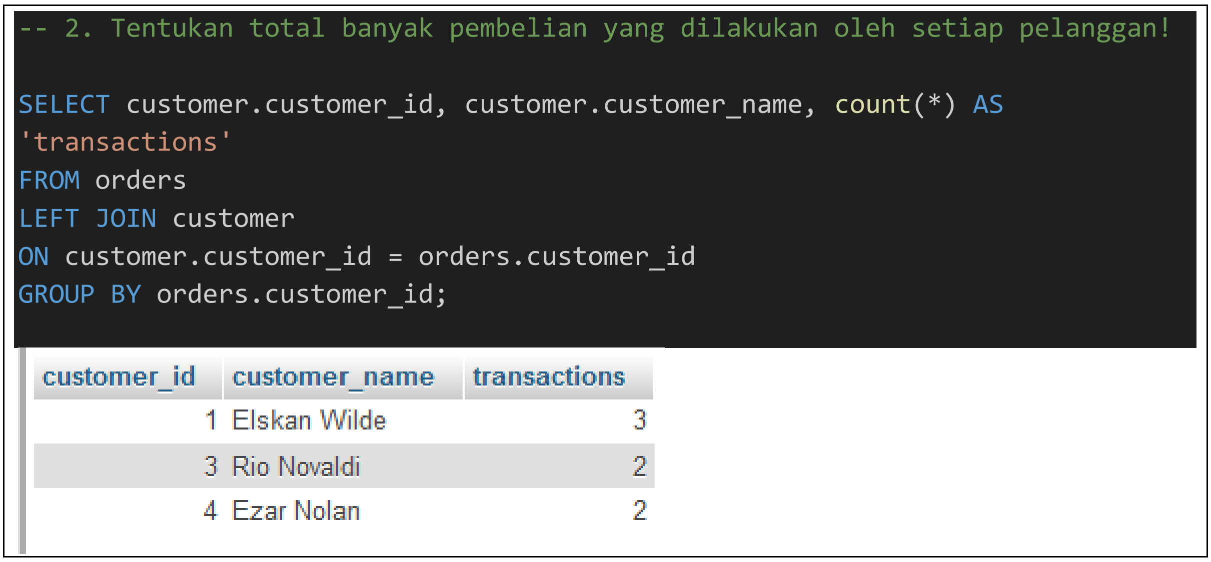
Nomor 1: Tentukanlah pelanggan yang tidak pernah membuat pesanan!



```
-- 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
);
```

customer_id	customer_name	customer_city
2	Aprilya Yusri	JAKARTA
5	Jessen Cornelius	BANDUNG

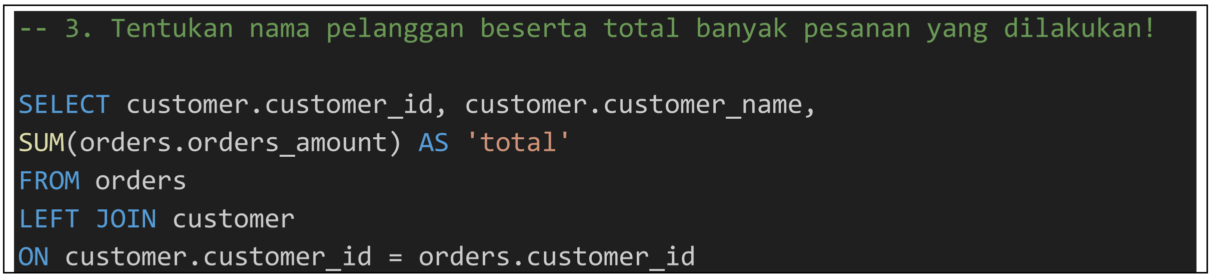
Nomor 2: Tentukan total banyak pembelian yang dilakukan oleh setiap pelanggan!



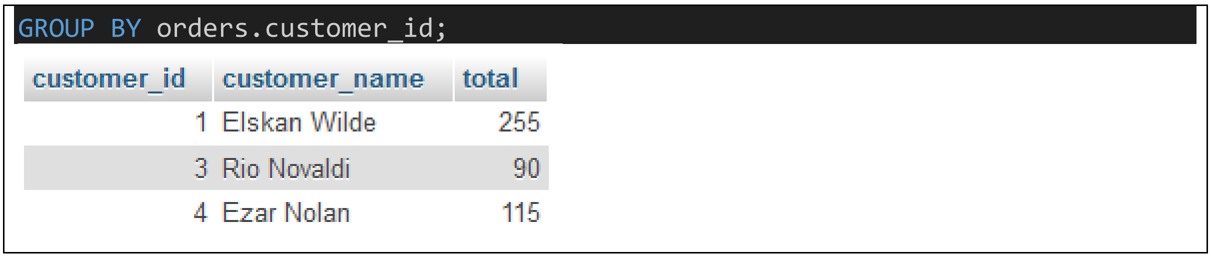
```
-- 2. Tentukan total banyak pembelian yang dilakukan oleh setiap pelanggan!

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;
```

customer_id	customer_name	transactions
1	Elskan Wilde	3
3	Rio Novaldi	2
4	Ezar Nolan	2

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



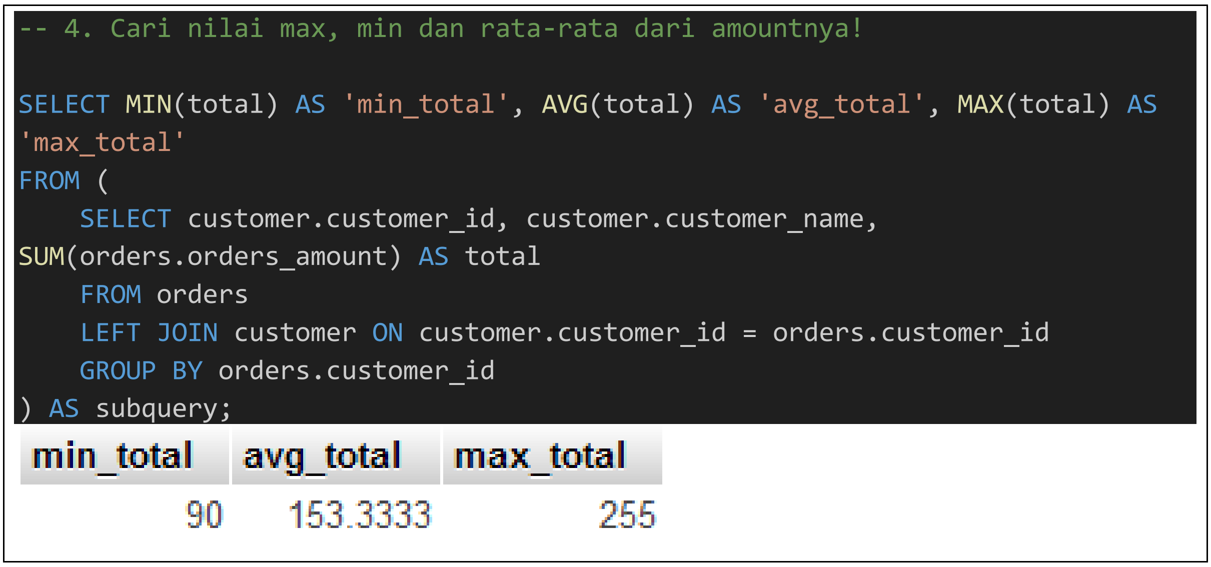
```
-- 3. Tentukan nama pelanggan beserta total banyak pesanan yang dilakukan!

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;
```

customer_id	customer_name	total
1	Elskan Wilde	255
3	Rio Novaldi	90
4	Ezar Nolan	115

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;
```

min_total	avg_total	max_total
90	153.3333	255