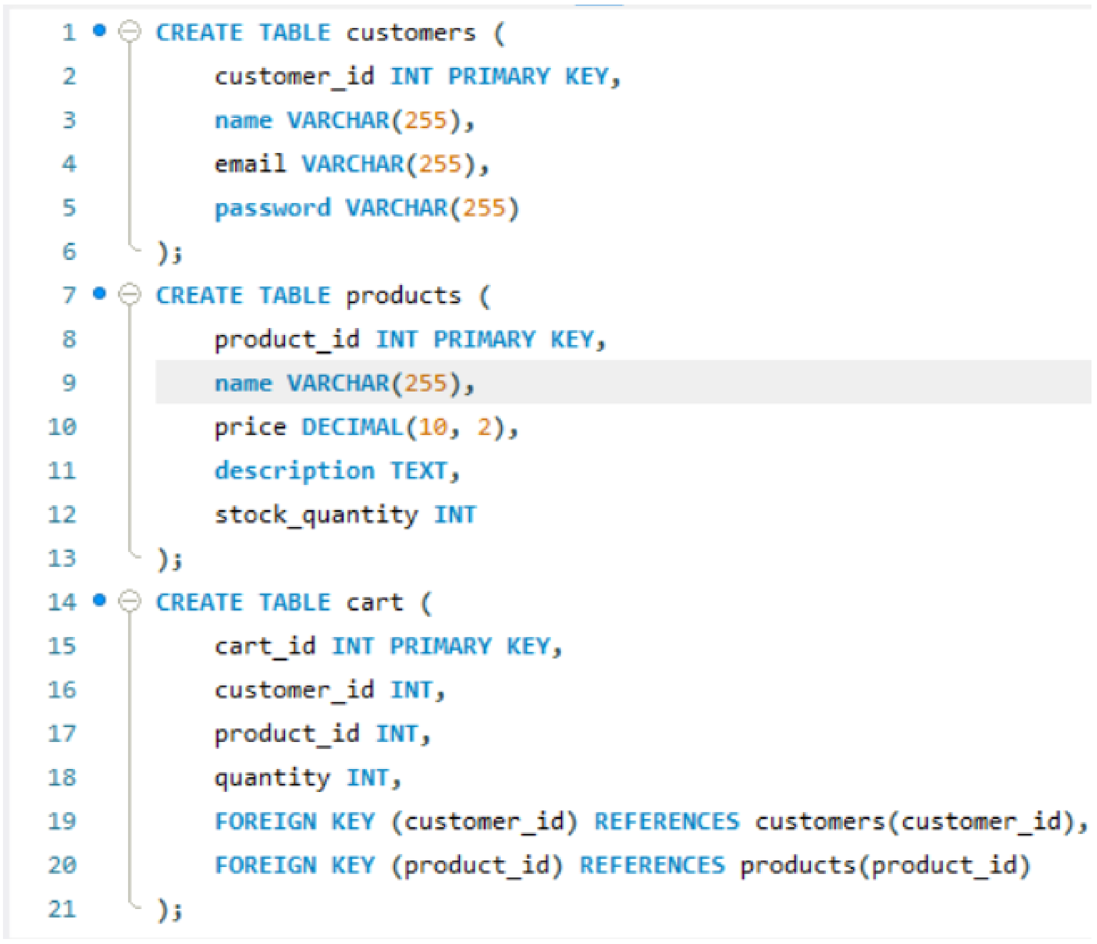
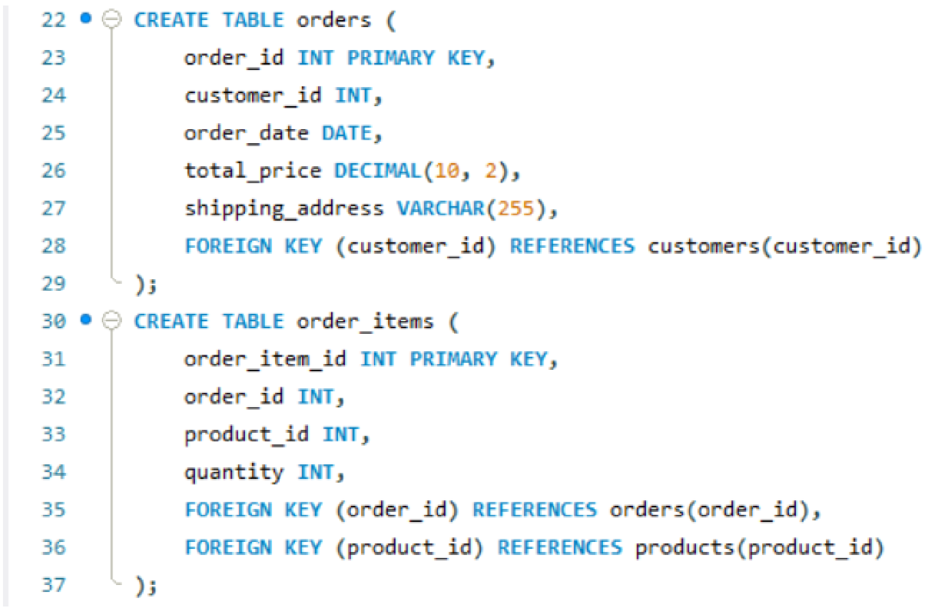
Ecommerce - SQL



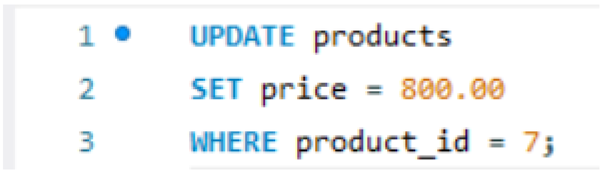
```sql
CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    name VARCHAR(255),
    email VARCHAR(255),
    password VARCHAR(255)
);
CREATE TABLE products (
    product_id INT PRIMARY KEY,
    name VARCHAR(255),
    price DECIMAL(10, 2),
    description TEXT,
    stock_quantity INT
);
CREATE TABLE cart (
    cart_id INT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    quantity INT,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```



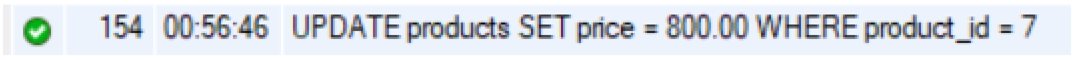
```sql
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    total_price DECIMAL(10, 2),
    shipping_address VARCHAR(255),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
CREATE TABLE order_items (
    order_item_id INT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT,
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

Queries

1. Update refrigerator product price to 800.

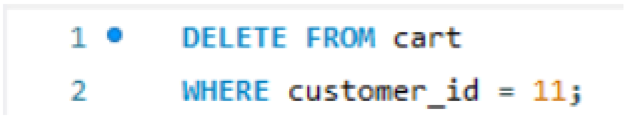


```
UPDATE products
SET price = 800.00
WHERE product_id = 7;
```



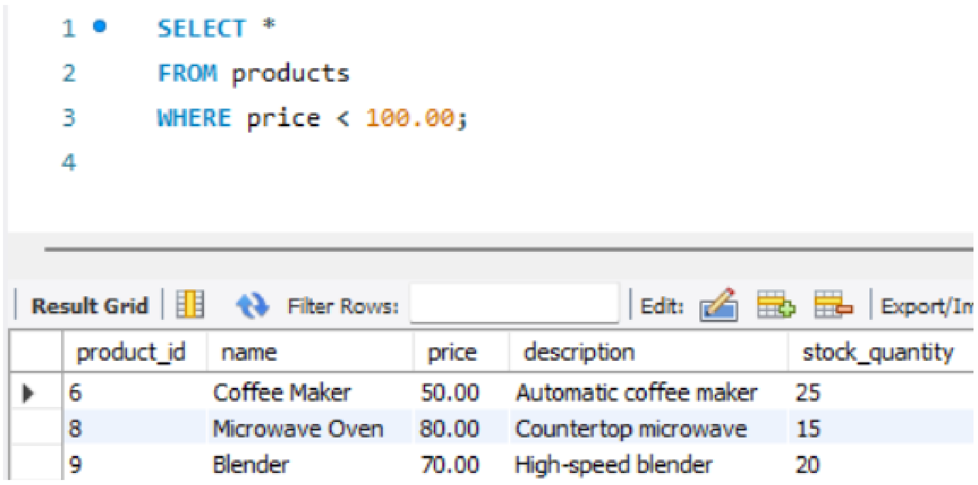
```
154 00:56:46 UPDATE products SET price = 800.00 WHERE product_id = 7
```

2. Remove all cart items for a specific customer.



```
DELETE FROM cart
WHERE customer_id = 11;
```

3. Retrieve Products Priced Below $100.

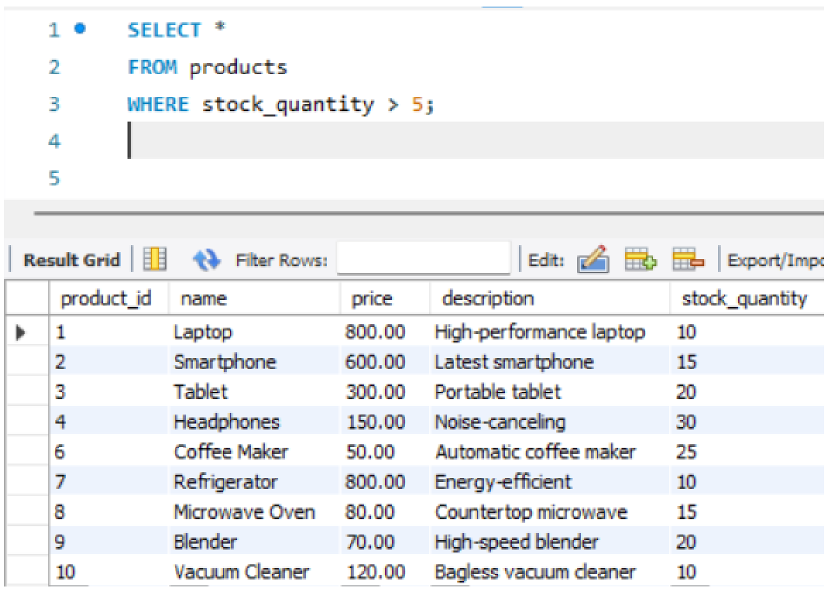


```
SELECT *
FROM products
WHERE price < 100.00;

```

	product_id	name	price	description	stock_quantity
▶	6	Coffee Maker	50.00	Automatic coffee maker	25
	8	Microwave Oven	80.00	Countertop microwave	15
	9	Blender	70.00	High-speed blender	20

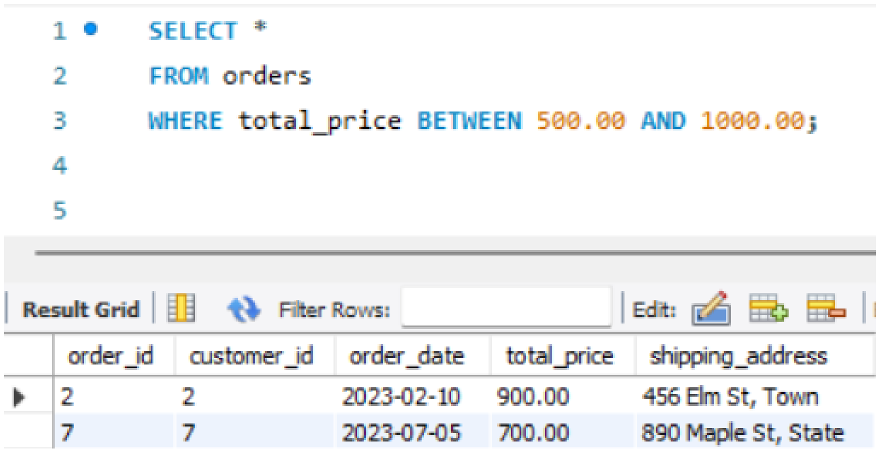
4. Find Products with Stock Quantity Greater Than 5.



```sql
SELECT *
FROM products
WHERE stock_quantity > 5;
```

product_id	name	price	description	stock_quantity
1	Laptop	800.00	High-performance laptop	10
2	Smartphone	600.00	Latest smartphone	15
3	Tablet	300.00	Portable tablet	20
4	Headphones	150.00	Noise-canceling	30
6	Coffee Maker	50.00	Automatic coffee maker	25
7	Refrigerator	800.00	Energy-efficient	10
8	Microwave Oven	80.00	Countertop microwave	15
9	Blender	70.00	High-speed blender	20
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10

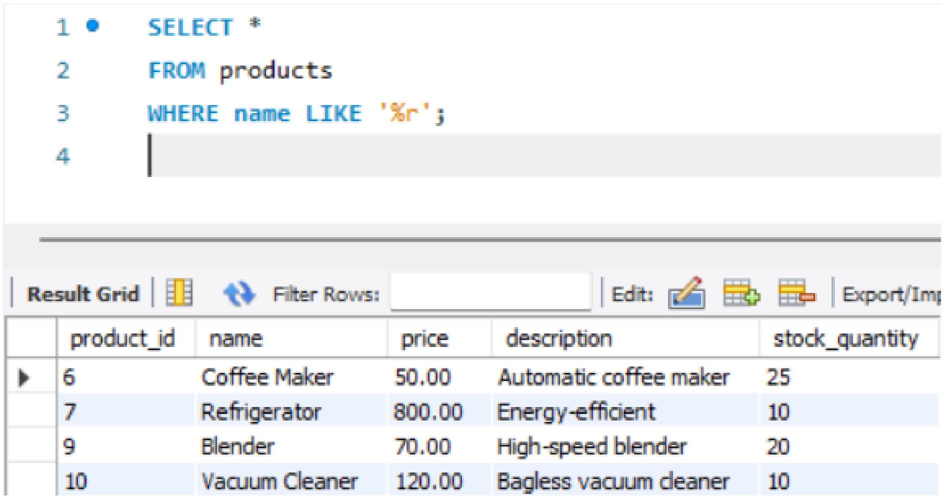
5. Retrieve Orders with Total Amount Between $500 and $1000.



```sql
SELECT *
FROM orders
WHERE total_price BETWEEN 500.00 AND 1000.00;
```

order_id	customer_id	order_date	total_price	shipping_address
2	2	2023-02-10	900.00	456 Elm St, Town
7	7	2023-07-05	700.00	890 Maple St, State

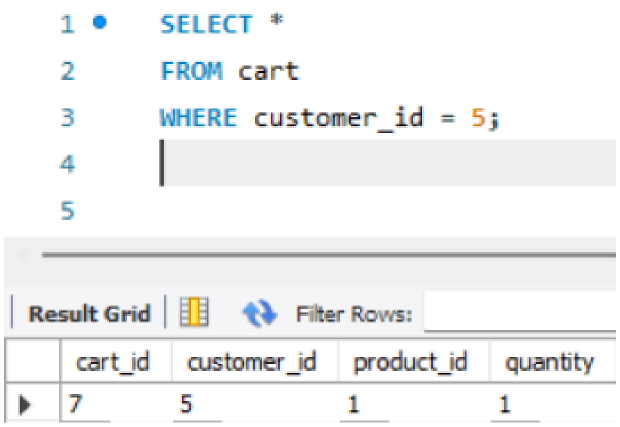
6. Find Products whose name ends with the letter ‘r’.



```sql
SELECT *
FROM products
WHERE name LIKE '%r';
```

product_id	name	price	description	stock_quantity
6	Coffee Maker	50.00	Automatic coffee maker	25
7	Refrigerator	800.00	Energy-efficient	10
9	Blender	70.00	High-speed blender	20
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10

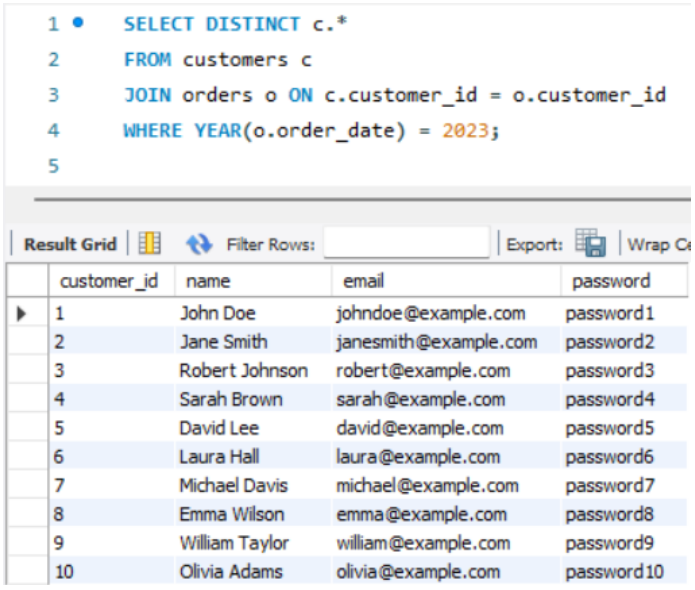
7. Retrieve Cart Items for Customer 5.



```sql
SELECT *
FROM cart
WHERE customer_id = 5;
```

cart_id	customer_id	product_id	quantity
7	5	1	1

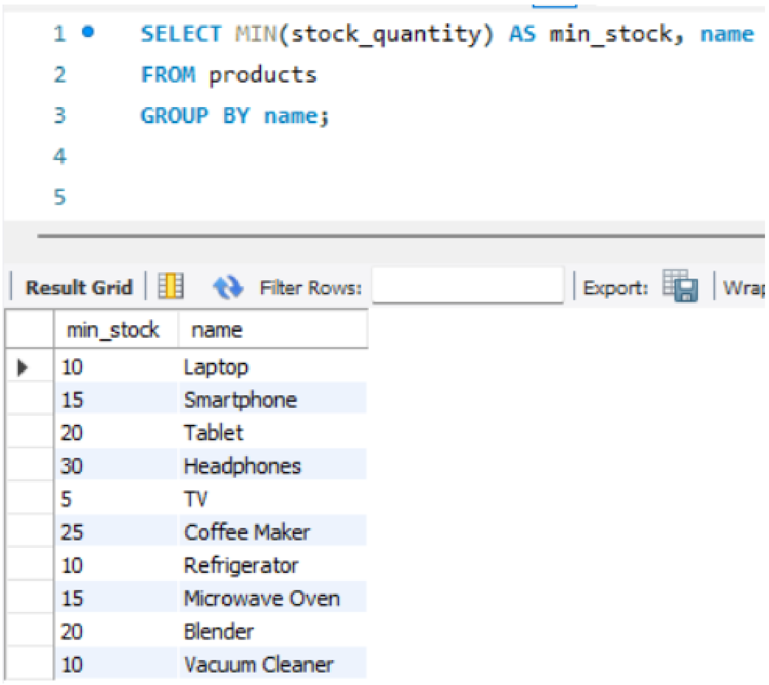
8. Find Customers Who Placed Orders in 2023.



```sql
SELECT DISTINCT c.*
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
WHERE YEAR(o.order_date) = 2023;
```

customer_id	name	email	password
1	John Doe	johndoe@example.com	password1
2	Jane Smith	janesmith@example.com	password2
3	Robert Johnson	robert@example.com	password3
4	Sarah Brown	sarah@example.com	password4
5	David Lee	david@example.com	password5
6	Laura Hall	laura@example.com	password6
7	Michael Davis	michael@example.com	password7
8	Emma Wilson	emma@example.com	password8
9	William Taylor	william@example.com	password9
10	Olivia Adams	olivia@example.com	password10

9. Determine the Minimum Stock Quantity for Each Product Category.

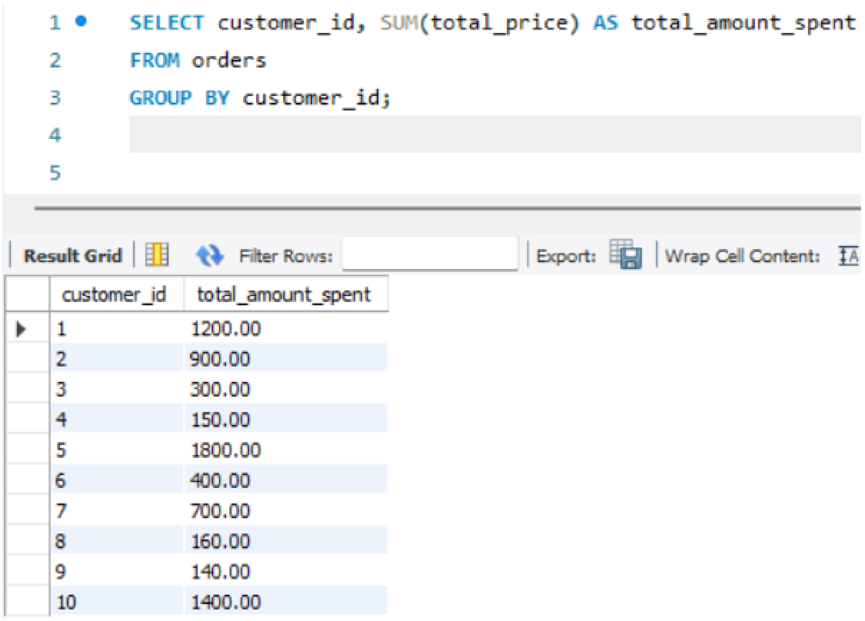


```
SELECT MIN(stock_quantity) AS min_stock, name
FROM products
GROUP BY name;
```

Result Grid | Filter Rows: | Export: | Wrap

min_stock	name
10	Laptop
15	Smartphone
20	Tablet
30	Headphones
5	TV
25	Coffee Maker
10	Refrigerator
15	Microwave Oven
20	Blender
10	Vacuum Cleaner

10. Calculate the Total Amount Spent by Each Customer.

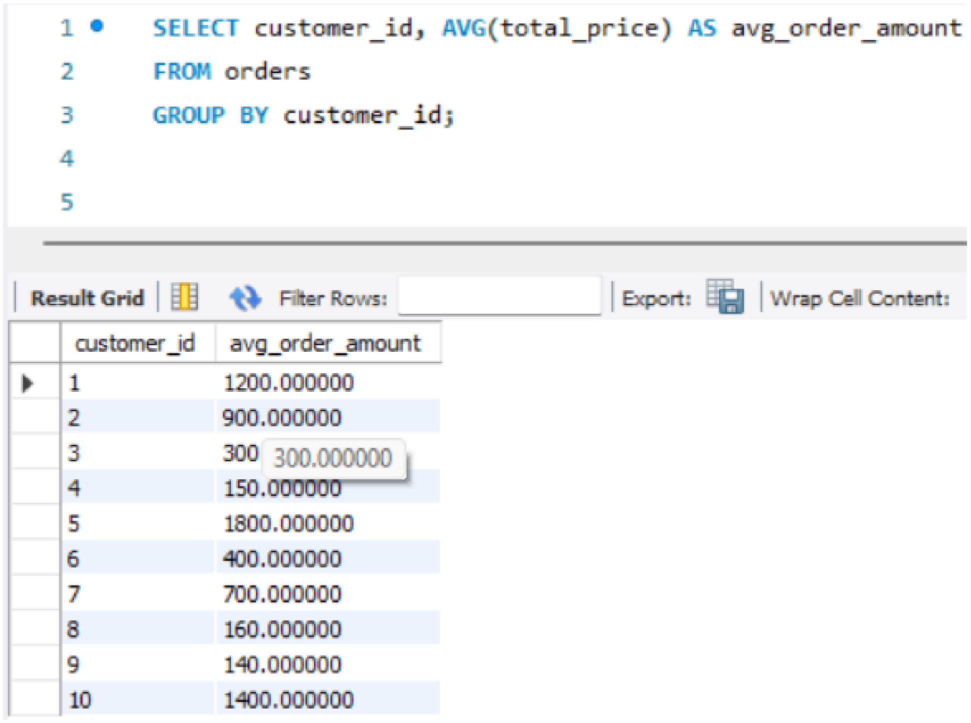


```
SELECT customer_id, SUM(total_price) AS total_amount_spent
FROM orders
GROUP BY customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	total_amount_spent
1	1200.00
2	900.00
3	300.00
4	150.00
5	1800.00
6	400.00
7	700.00
8	160.00
9	140.00
10	1400.00

11. Find the Average Order Amount for Each Customer.

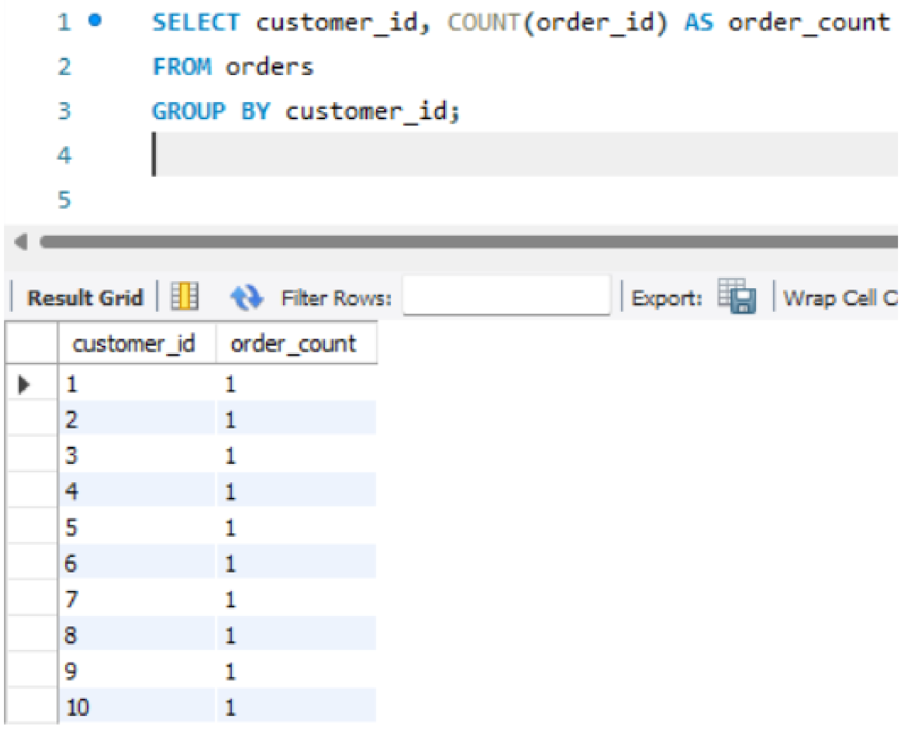


```sql
SELECT customer_id, AVG(total_price) AS avg_order_amount
FROM orders
GROUP BY customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	avg_order_amount
1	1200.000000
2	900.000000
3	300.000000
4	150.000000
5	1800.000000
6	400.000000
7	700.000000
8	160.000000
9	140.000000
10	1400.000000

12. Count the Number of Orders Placed by Each Customer.

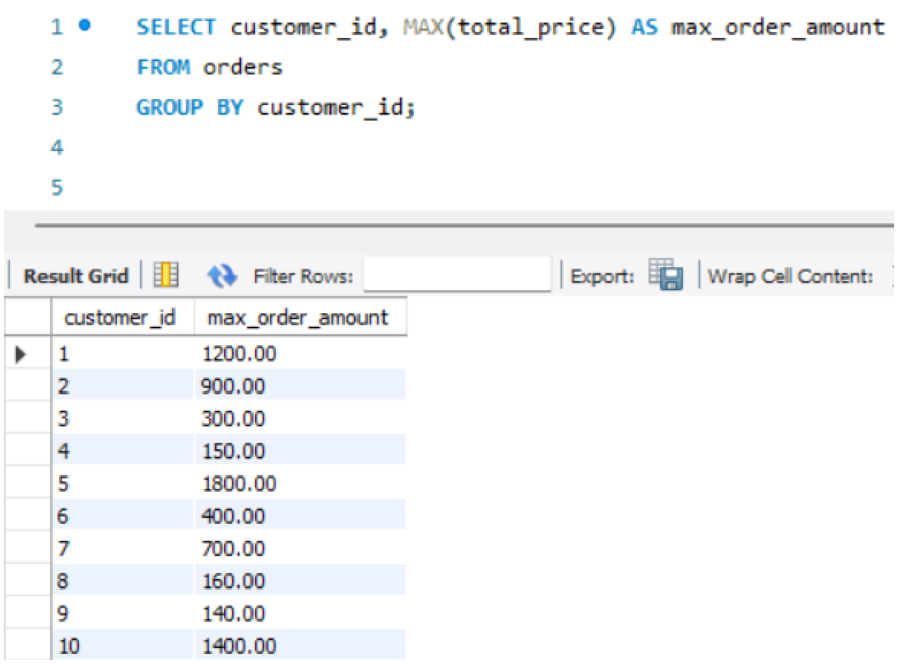


```sql
SELECT customer_id, COUNT(order_id) AS order_count
FROM orders
GROUP BY customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

customer_id	order_count
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

13. Find the Maximum Order Amount for Each Customer.

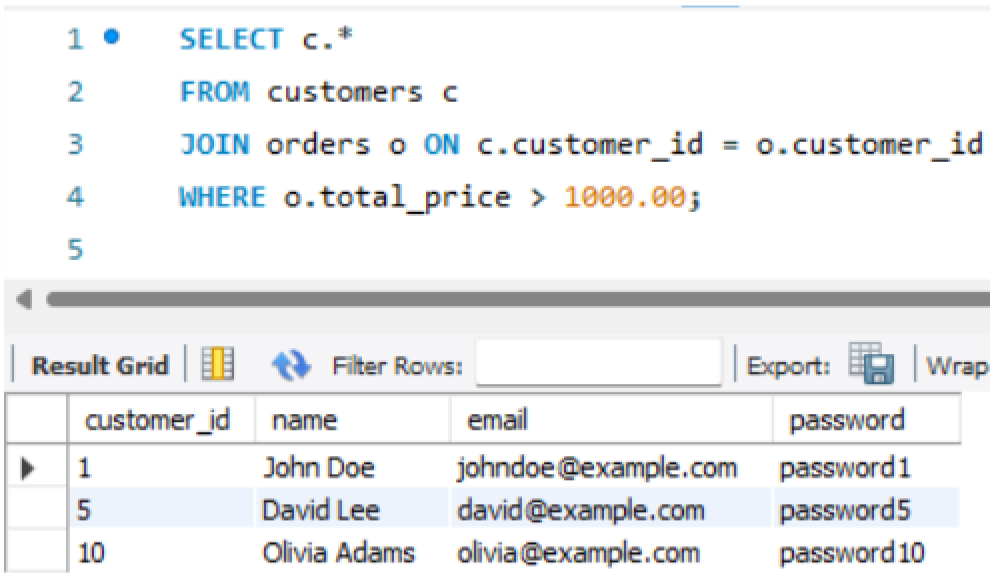


```sql
SELECT customer_id, MAX(total_price) AS max_order_amount
FROM orders
GROUP BY customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	max_order_amount
1	1200.00
2	900.00
3	300.00
4	150.00
5	1800.00
6	400.00
7	700.00
8	160.00
9	140.00
10	1400.00

14. Get Customers Who Placed Orders Totaling Over $1000.

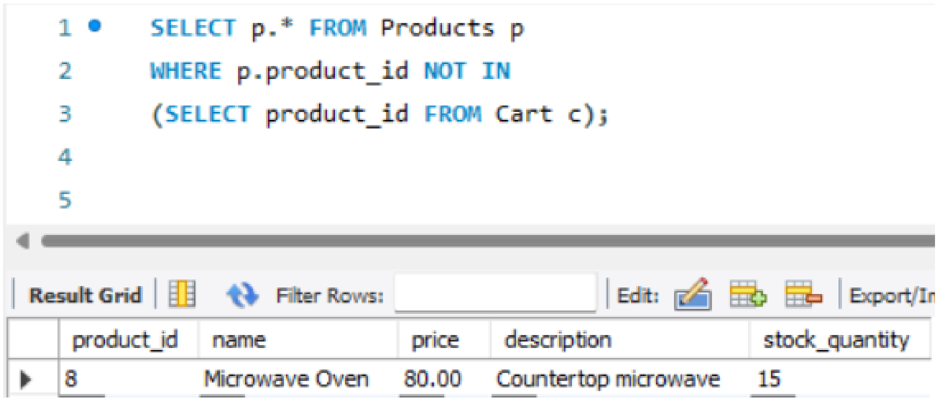


```sql
SELECT c.*
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
WHERE o.total_price > 1000.00;
```

Result Grid | Filter Rows: | Export: | Wrap

customer_id	name	email	password
1	John Doe	johndoe@example.com	password1
5	David Lee	david@example.com	password5
10	Olivia Adams	olivia@example.com	password10

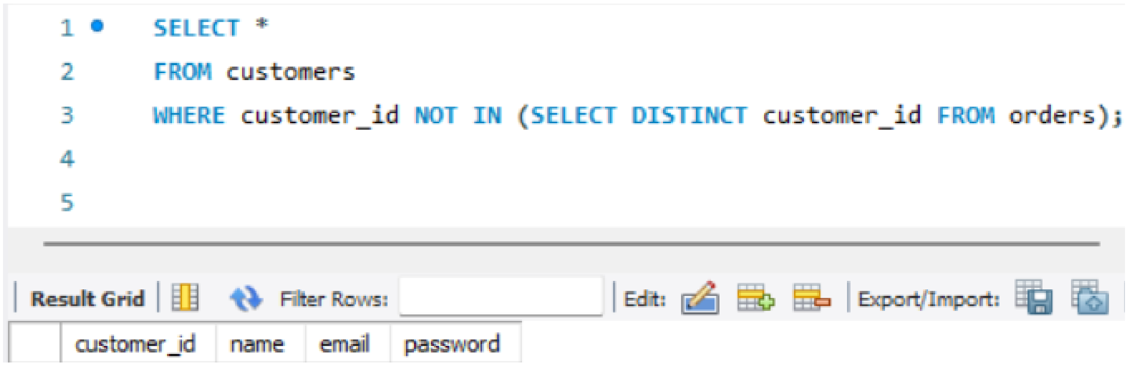
15. Subquery to Find Products Not in the Cart.



```sql
SELECT p.* FROM Products p
WHERE p.product_id NOT IN
(SELECT product_id FROM Cart c);
```

product_id	name	price	description	stock_quantity
8	Microwave Oven	80.00	Countertop microwave	15

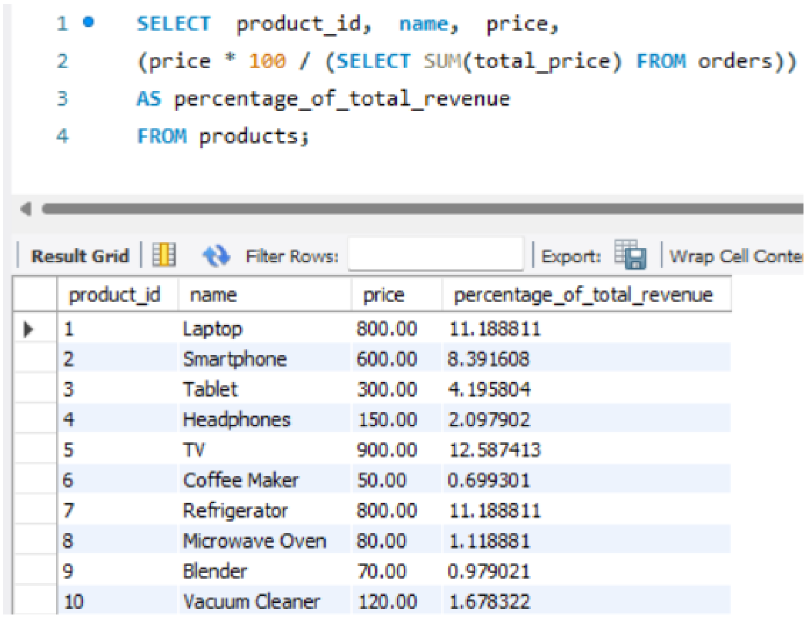
16. Subquery to Find Customers Who Haven't Placed Orders.



```sql
SELECT *
FROM customers
WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM orders);
```

customer_id	name	email	password

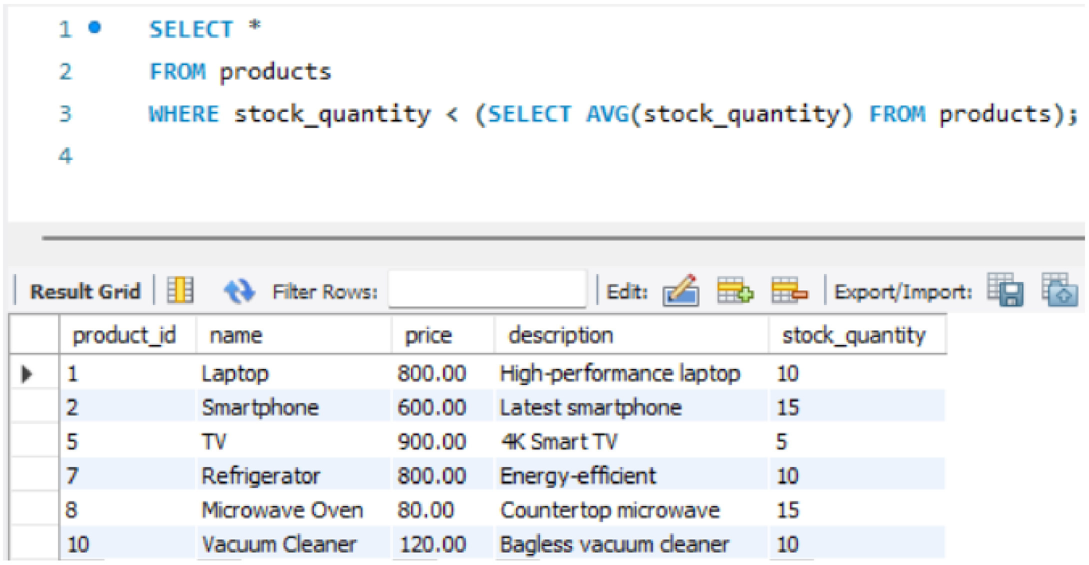
17. Subquery to Calculate the Percentage of Total Revenue for a Product.



```sql
SELECT  product_id,  name,  price,
(price * 100 / (SELECT SUM(total_price) FROM orders))
AS percentage_of_total_revenue
FROM products;
```

product_id	name	price	percentage_of_total_revenue
1	Laptop	800.00	11.188811
2	Smartphone	600.00	8.391608
3	Tablet	300.00	4.195804
4	Headphones	150.00	2.097902
5	TV	900.00	12.587413
6	Coffee Maker	50.00	0.699301
7	Refrigerator	800.00	11.188811
8	Microwave Oven	80.00	1.118881
9	Blender	70.00	0.979021
10	Vacuum Cleaner	120.00	1.678322

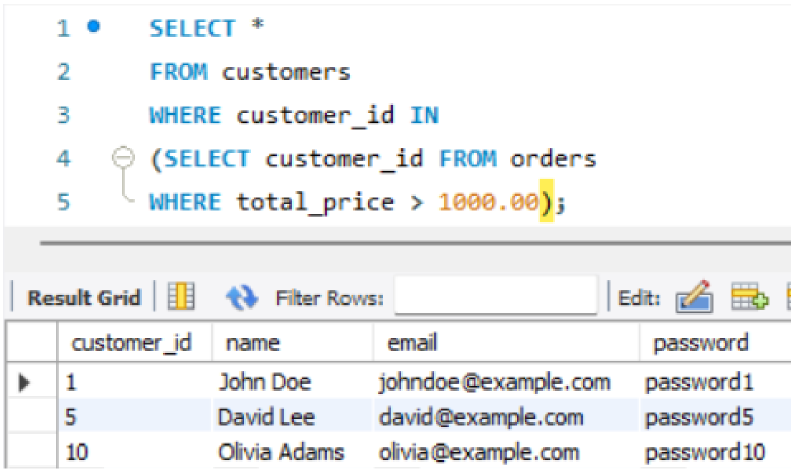
18. Subquery to Find Products with Low Stock.



```
SELECT *
FROM products
WHERE stock_quantity < (SELECT AVG(stock_quantity) FROM products);
```

product_id	name	price	description	stock_quantity
1	Laptop	800.00	High-performance laptop	10
2	Smartphone	600.00	Latest smartphone	15
5	TV	900.00	4K Smart TV	5
7	Refrigerator	800.00	Energy-efficient	10
8	Microwave Oven	80.00	Countertop microwave	15
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10

19. Subquery to Find Customers Who Placed High-Value Orders.



```
SELECT *
FROM customers
WHERE customer_id IN
(SELECT customer_id FROM orders
WHERE total_price > 1000.00);
```

customer_id	name	email	password
1	John Doe	johndoe@example.com	password1
5	David Lee	david@example.com	password5
10	Olivia Adams	olivia@example.com	password10