**SQL CODING CHALLENGE**

**Ecommerce – SQL**

* **PERSIS FANNY ELOITE M**

mysql> create database ecommerce;

mysql> use ecommerce;

Database changed

mysql> CREATE TABLE customers (

-> customer\_id INT PRIMARY KEY,

-> first\_name VARCHAR(50) NOT NULL,

-> last\_name VARCHAR(50) NOT NULL,

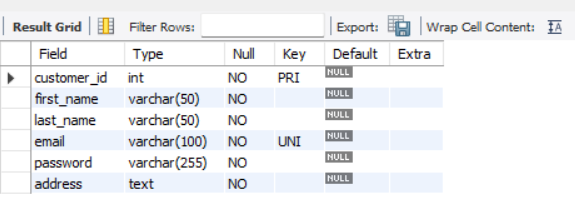
-> email VARCHAR(100) UNIQUE NOT NULL,

-> password VARCHAR(255) NOT NULL,

-> address TEXT NOT NULL

-> );

Query OK, 0 rows affected (0.05 sec)



mysql> CREATE TABLE products (

-> product\_id INT PRIMARY KEY,

-> name VARCHAR(100) NOT NULL,

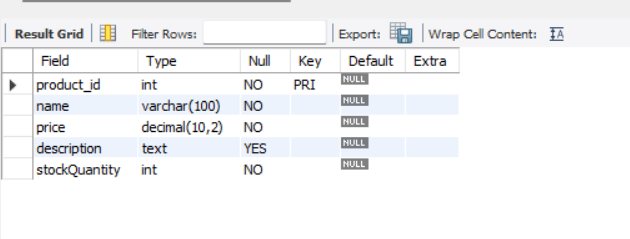
-> price DECIMAL(10,2) NOT NULL,

-> description TEXT,

-> stockQuantity INT NOT NULL

-> );

Query OK, 0 rows affected (0.02 sec)



mysql> CREATE TABLE cart (

-> cart\_id INT PRIMARY KEY,

-> customer\_id INT NOT NULL,

-> product\_id INT NOT NULL,

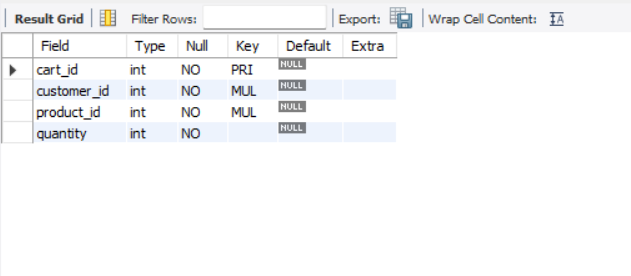
-> quantity INT NOT NULL,

-> FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id) ON DELETE CASCADE,

-> FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

-> );

Query OK, 0 rows affected (0.04 sec)



mysql> CREATE TABLE orders (

-> order\_id INT PRIMARY KEY,

-> customer\_id INT NOT NULL,

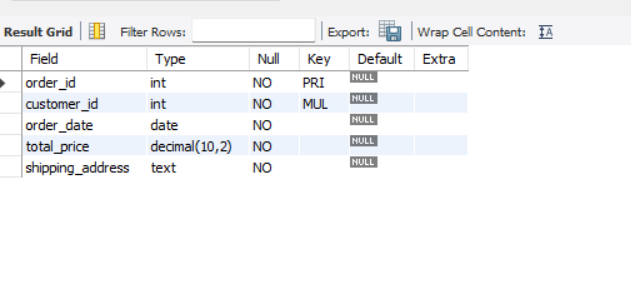
-> order\_date DATE NOT NULL,

-> total\_price DECIMAL(10,2) NOT NULL,

-> shipping\_address TEXT NOT NULL,

-> FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id) ON DELETE CASCADE

-> );



mysql> CREATE TABLE order\_items (

-> order\_item\_id INT PRIMARY KEY,

-> order\_id INT NOT NULL,

-> product\_id INT NOT NULL,

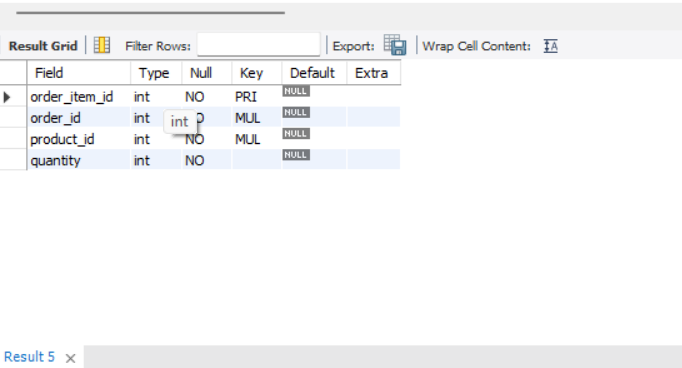
-> quantity INT NOT NULL,

-> FOREIGN KEY (order\_id) REFERENCES orders(order\_id) ON DELETE CASCADE,

-> FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

-> );

Query OK, 0 rows affected (0.04 sec)



4 rows in set (0.00 sec)

mysql> INSERT INTO customers (customer\_id, first\_name, last\_name, email, password, address) VALUES

-> (1, 'Alice', 'Johnson', 'alice@example.com', 'password123', '123 Main St, NY'),

-> (2, 'Bob', 'Smith', 'bob@example.com', 'securepass', '456 Elm St, CA'),

-> (3, 'Charlie', 'Brown', 'charlie@example.com', 'charliepass', '789 Oak St, TX'),

-> (4, 'David', 'Williams', 'david@example.com', 'davidpass', '321 Pine St, FL'),

-> (5, 'Eve', 'Miller', 'eve@example.com', 'evepass', '654 Maple St, WA'),

-> (6, 'Frank', 'Anderson', 'frank@example.com', 'frankpass', '987 Cedar St, IL'),

-> (7, 'Grace', 'Moore', 'grace@example.com', 'gracepass', '741 Birch St, OH'),

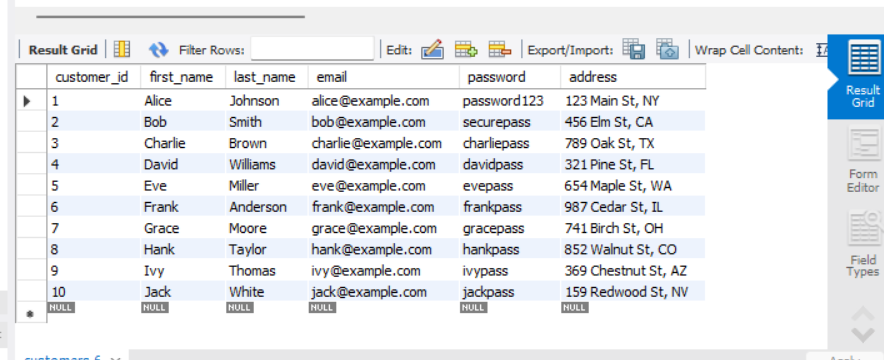
-> (8, 'Hank', 'Taylor', 'hank@example.com', 'hankpass', '852 Walnut St, CO'),

-> (9, 'Ivy', 'Thomas', 'ivy@example.com', 'ivypass', '369 Chestnut St, AZ'),

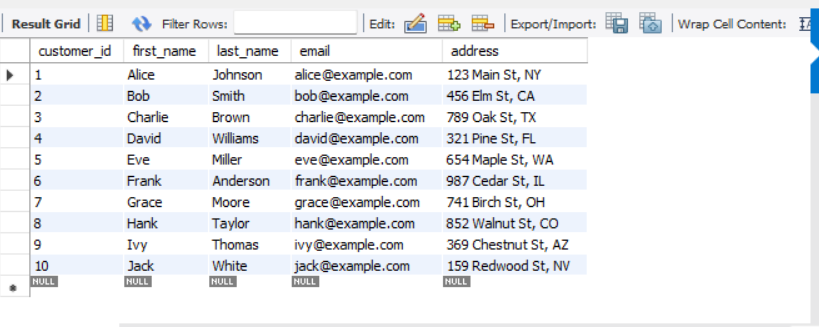
-> (10, 'Jack', 'White', 'jack@example.com', 'jackpass', '159 Redwood St, NV');

Query OK, 10 rows affected (0.01 sec)

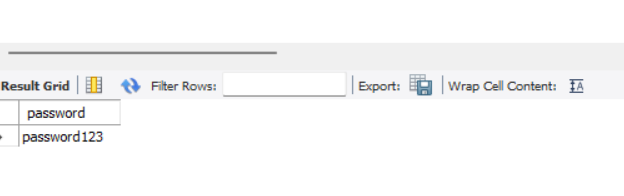
Records: 10 Duplicates: 0 Warnings: 0



mysql> SELECT customer\_id, first\_name, last\_name, email, address FROM customers;



SELECT password FROM customers WHERE email = 'alice@example.com';



mysql> INSERT INTO products (product\_id, name, price, description, stockQuantity) VALUES

-> (1, 'Refrigerator', 750.00, 'Double-door fridge', 10),

-> (2, 'Laptop', 1200.00, 'Gaming laptop', 5),

-> (3, 'Vacuum Cleaner', 150.00, 'Robotic vacuum cleaner', 8),

-> (4, 'Microwave', 200.00, 'Compact microwave oven', 7),

-> (5, 'Smartphone', 999.00, 'Latest model smartphone', 15),

-> (6, 'Headphones', 120.00, 'Noise-canceling headphones', 20),

-> (7, 'Smart TV', 1100.00, '4K UHD Smart TV', 6),

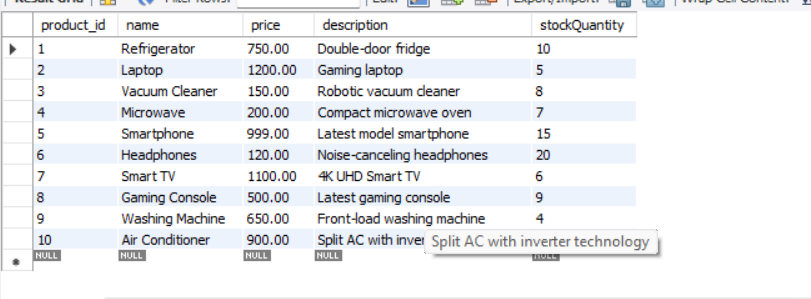
-> (8, 'Gaming Console', 500.00, 'Latest gaming console', 9),

-> (9, 'Washing Machine', 650.00, 'Front-load washing machine', 4),

-> (10, 'Air Conditioner', 900.00, 'Split AC with inverter technology', 5);

Query OK, 10 rows affected (0.01 sec)

Records: 10 Duplicates: 0 Warnings: 0



mysql> INSERT INTO cart (cart\_id, customer\_id, product\_id, quantity) VALUES

-> (1, 1, 2, 1),

-> (2, 2, 3, 2),

-> (3, 3, 5, 1),

-> (4, 4, 6, 2),

-> (5, 5, 8, 1),

-> (6, 6, 1, 1),

-> (7, 7, 4, 1),

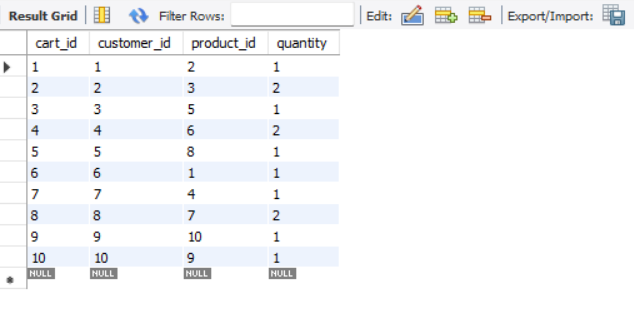
-> (8, 8, 7, 2),

-> (9, 9, 10, 1),

-> (10, 10, 9, 1);

Query OK, 10 rows affected (0.01 sec)

Records: 10 Duplicates: 0 Warnings: 0



mysql> INSERT INTO orders (order\_id, customer\_id, order\_date, total\_price, shipping\_address) VALUES

-> (1, 1, '2023-03-15', 1200.00, '123 Main St, NY'),

-> (2, 2, '2023-06-21', 300.00, '456 Elm St, CA'),

-> (3, 3, '2023-07-19', 999.00, '789 Oak St, TX'),

-> (4, 4, '2023-08-22', 500.00, '321 Pine St, FL'),

-> (5, 5, '2023-09-30', 650.00, '654 Maple St, WA'),

-> (6, 6, '2023-10-05', 750.00, '987 Cedar St, IL'),

-> (7, 7, '2023-11-10', 1100.00, '741 Birch St, OH'),

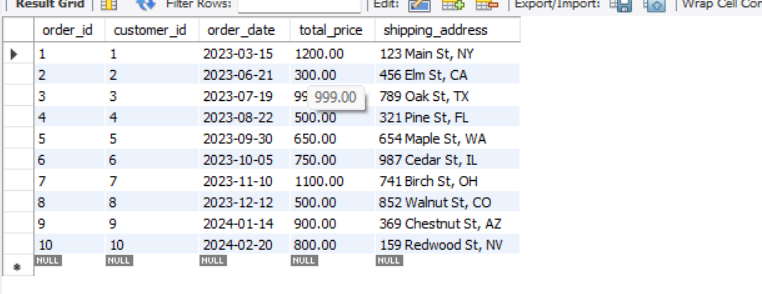
-> (8, 8, '2023-12-12', 500.00, '852 Walnut St, CO'),

-> (9, 9, '2024-01-14', 900.00, '369 Chestnut St, AZ'),

-> (10, 10, '2024-02-20', 800.00, '159 Redwood St, NV');

Query OK, 10 rows affected (0.01 sec)

Records: 10 Duplicates: 0 Warnings: 0



mysql> INSERT INTO order\_items (order\_item\_id, order\_id, product\_id, quantity) VALUES

-> (1, 1, 2, 1),

-> (2, 2, 3, 2),

-> (3, 3, 5, 1),

-> (4, 4, 6, 2),

-> (5, 5, 9, 1),

-> (6, 6, 1, 1),

-> (7, 7, 4, 1),

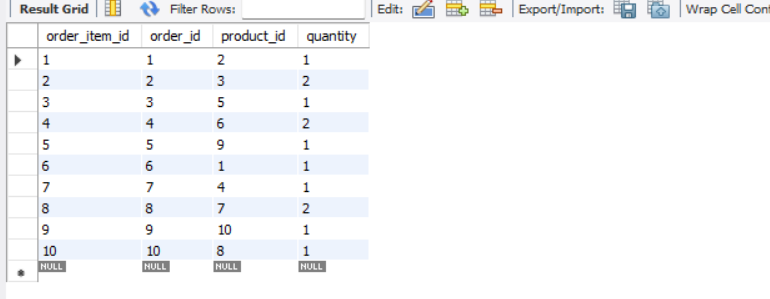
-> (8, 8, 7, 2),

-> (9, 9, 10, 1),

-> (10, 10, 8, 1);

Query OK, 10 rows affected (0.01 sec)

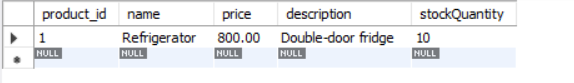
Records: 10 Duplicates: 0 Warnings: 0



**QUESTIONS:**

**1. Update refrigerator product price to 800**

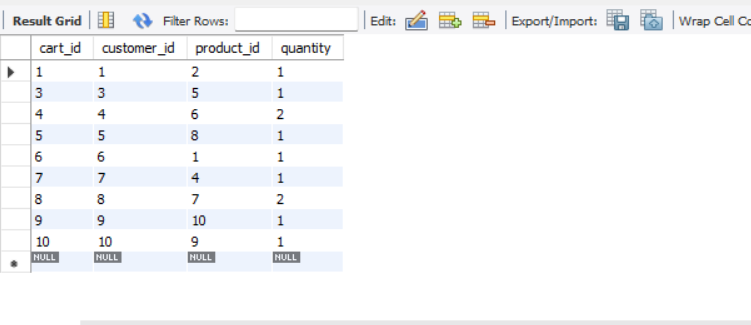
UPDATE products SET price = 800 WHERE name = 'Refrigerator';



**2. Remove all cart items for a specific customer**

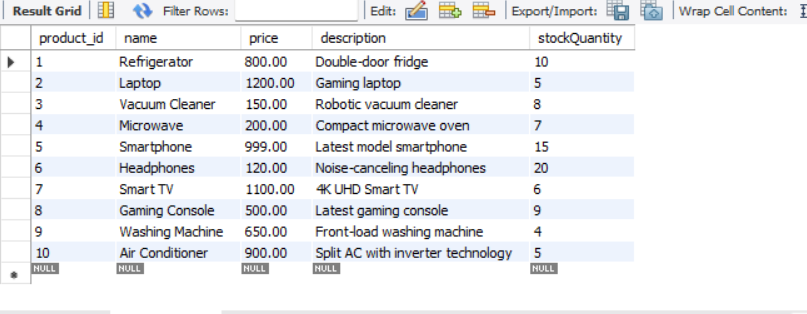
DELETE FROM cart WHERE customer\_id = 2;

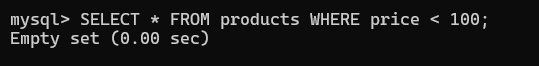
select \* from cart;



**3. Retrieve Products Priced Below $100**

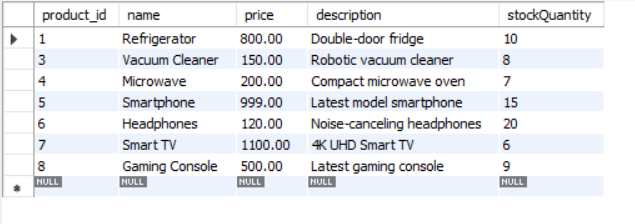
SELECT \* FROM products WHERE price < 100;





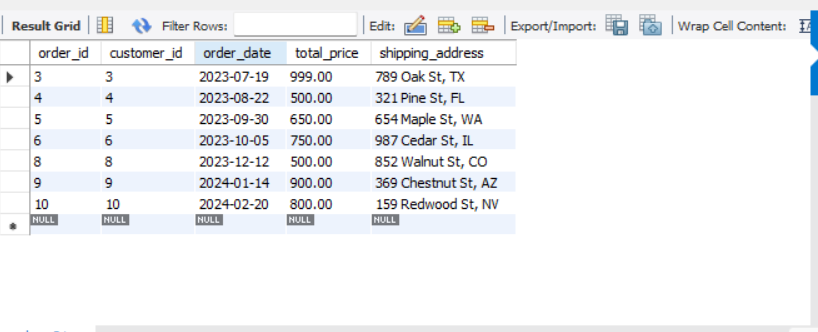
**4. Find Products with Stock Quantity Greater Than 5**

SELECT \* FROM products WHERE stockQuantity > 5;



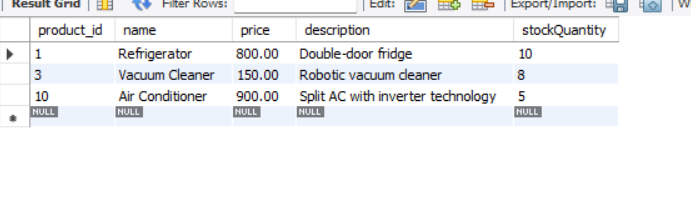
**5. Retrieve Orders with Total Amount Between $500 and $1000**

SELECT \* FROM orders WHERE total\_price BETWEEN 500 AND 1000;



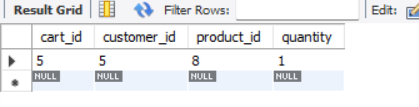
**6. Find Products which name end with letter ‘r’**

SELECT \* FROM products WHERE name LIKE '%r';



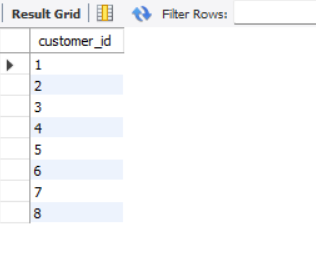
**7. Retrieve Cart Items for Customer 5**

SELECT \* FROM cart WHERE customer\_id = 5;



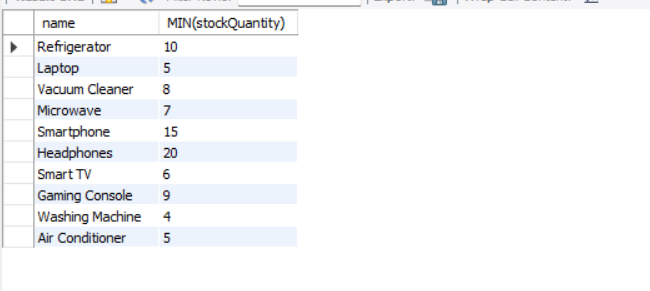
**8. Find Customers Who Placed Orders in 2023**

SELECT DISTINCT customer\_id FROM orders WHERE YEAR(order\_date) = 2023;



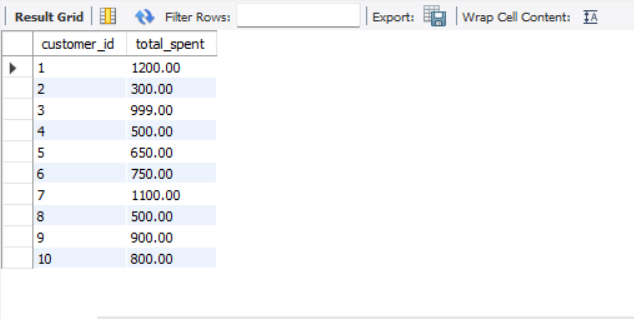
**9. Determine the Minimum Stock Quantity for Each Product Category**

SELECT name, MIN(stockQuantity) FROM products GROUP BY name;



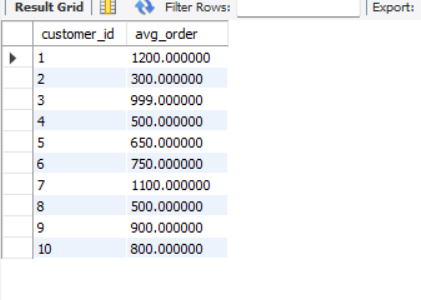
**10. Calculate the Total Amount Spent by Each Customer**

SELECT customer\_id, SUM(total\_price) AS total\_spent FROM orders GROUP BY customer\_id;



**11. Find the Average Order Amount for Each Customer**

SELECT customer\_id, AVG(total\_price) AS avg\_order FROM orders GROUP BY customer\_id;



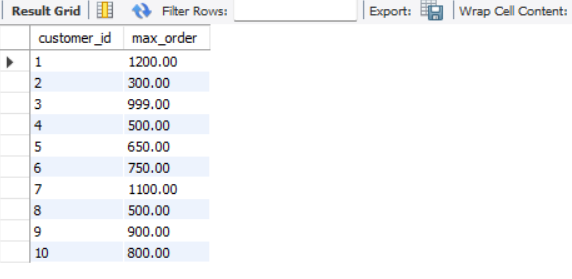
**12. Count the Number of Orders Placed by Each Customer**

SELECT customer\_id, COUNT(\*) AS order\_count FROM orders GROUP BY customer\_id;



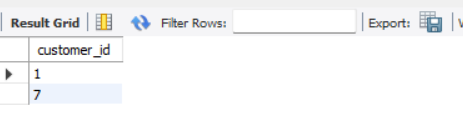
**13. Find the Maximum Order Amount for Each Customer**

SELECT customer\_id, MAX(total\_price) AS max\_order FROM orders GROUP BY customer\_id;



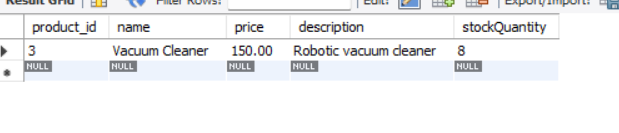
**14. Get Customers Who Placed Orders Totaling Over $1000**

SELECT customer\_id FROM orders GROUP BY customer\_id HAVING SUM(total\_price) > 1000;



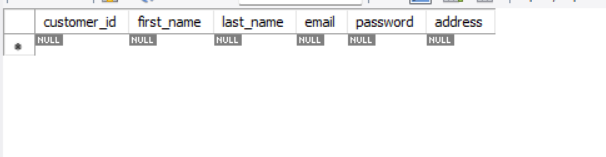
**15. Subquery to Find Products Not in the Cart**

SELECT \* FROM products WHERE product\_id NOT IN (SELECT product\_id FROM cart);



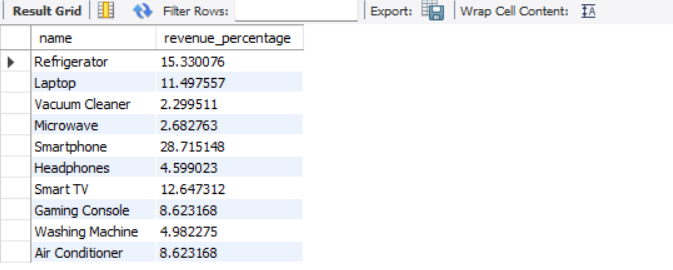
**16. Subquery to Find Customers Who Haven't Placed Orders**

SELECT \* FROM customers WHERE customer\_id NOT IN (SELECT customer\_id FROM orders);



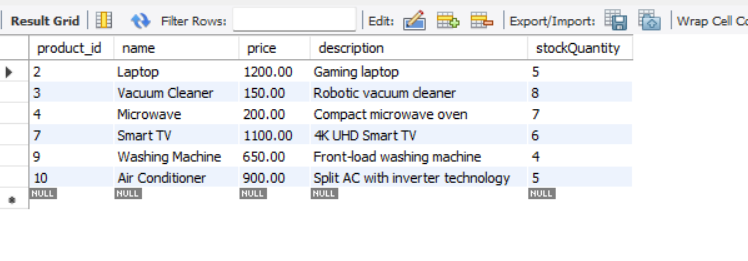
**17. Subquery to Calculate the Percentage of Total Revenue for a Product**

SELECT name, (price \* stockQuantity) / (SELECT SUM(price \* stockQuantity) FROM products) \* 100 AS revenue\_percentage FROM products;



**18. Subquery to Find Products with Low Stock**

SELECT \* FROM products WHERE stockQuantity < (SELECT AVG(stockQuantity) FROM products);



**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 > (SELECT AVG(total\_price) FROM orders));

