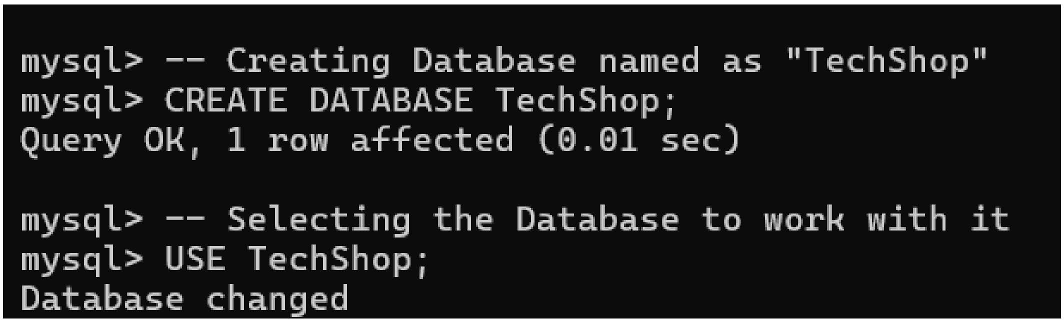
ASSIGNMENT 1 – SQL

ELECTRONIC GADGETS

Swathi Baskaran [Batch 4]

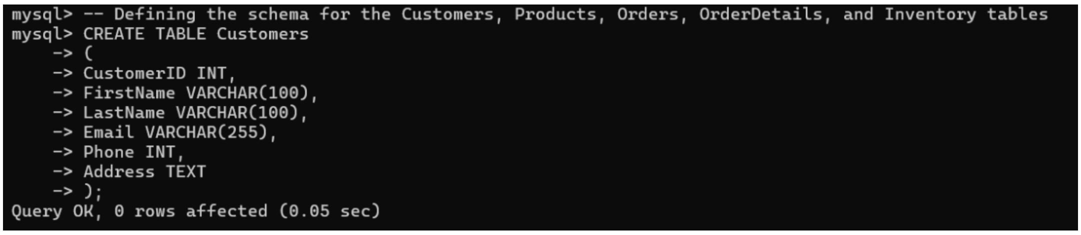
Task 1: Database Design

1. Create the database named "TechShop"

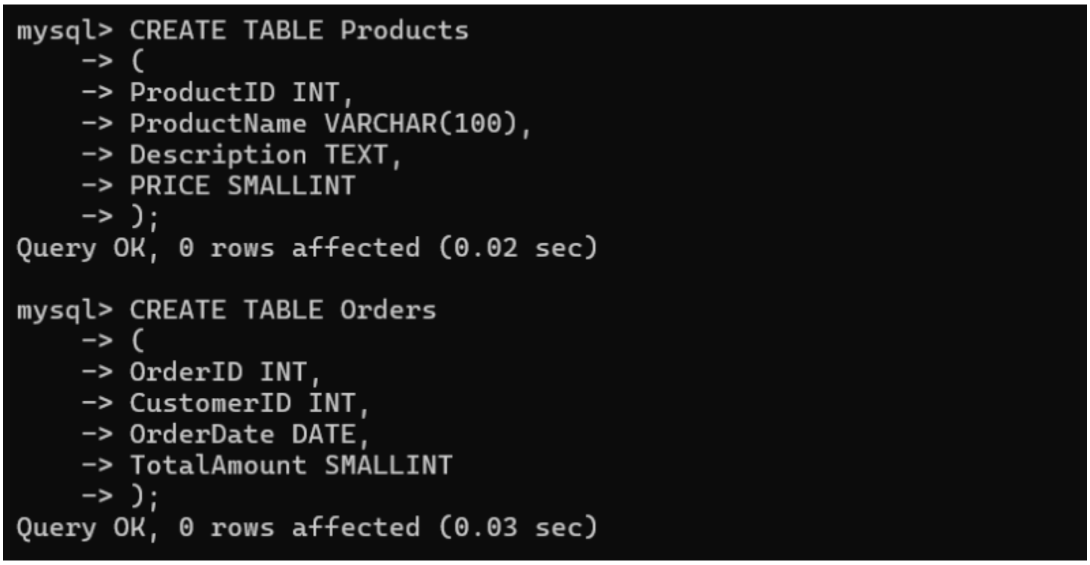


```
mysql> -- Creating Database named as "TechShop"
mysql> CREATE DATABASE TechShop;
Query OK, 1 row affected (0.01 sec)

mysql> -- Selecting the Database to work with it
mysql> USE TechShop;
Database changed
```

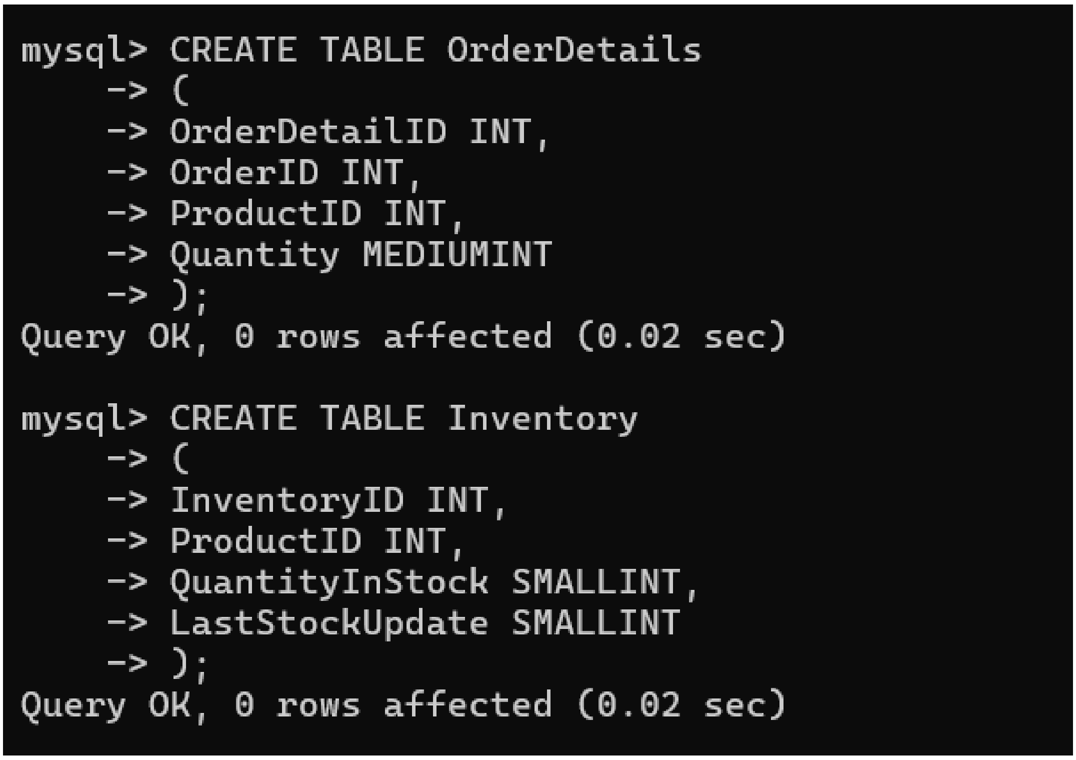
2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.



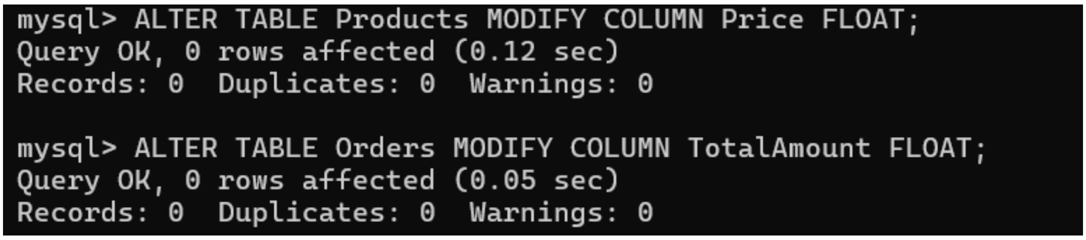
```
mysql> -- Defining the schema for the Customers, Products, Orders, OrderDetails, and Inventory tables
mysql> CREATE TABLE Customers
    -> (
    -> CustomerID INT,
    -> FirstName VARCHAR(100),
    -> LastName VARCHAR(100),
    -> Email VARCHAR(255),
    -> Phone INT,
    -> Address TEXT
    -> );
Query OK, 0 rows affected (0.05 sec)
```



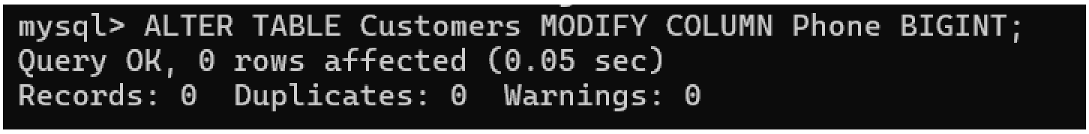
```
mysql> CREATE TABLE Products
    -> (
    -> ProductID INT,
    -> ProductName VARCHAR(100),
    -> Description TEXT,
    -> PRICE SMALLINT
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Orders
    -> (
    -> OrderID INT,
    -> CustomerID INT,
    -> OrderDate DATE,
    -> TotalAmount SMALLINT
    -> );
Query OK, 0 rows affected (0.03 sec)
```



```
mysql> CREATE TABLE OrderDetails
    -> (
    -> OrderDetailID INT,
    -> OrderID INT,
    -> ProductID INT,
    -> Quantity MEDIUMINT
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Inventory
    -> (
    -> InventoryID INT,
    -> ProductID INT,
    -> QuantityInStock SMALLINT,
    -> LastStockUpdate SMALLINT
    -> );
Query OK, 0 rows affected (0.02 sec)
```

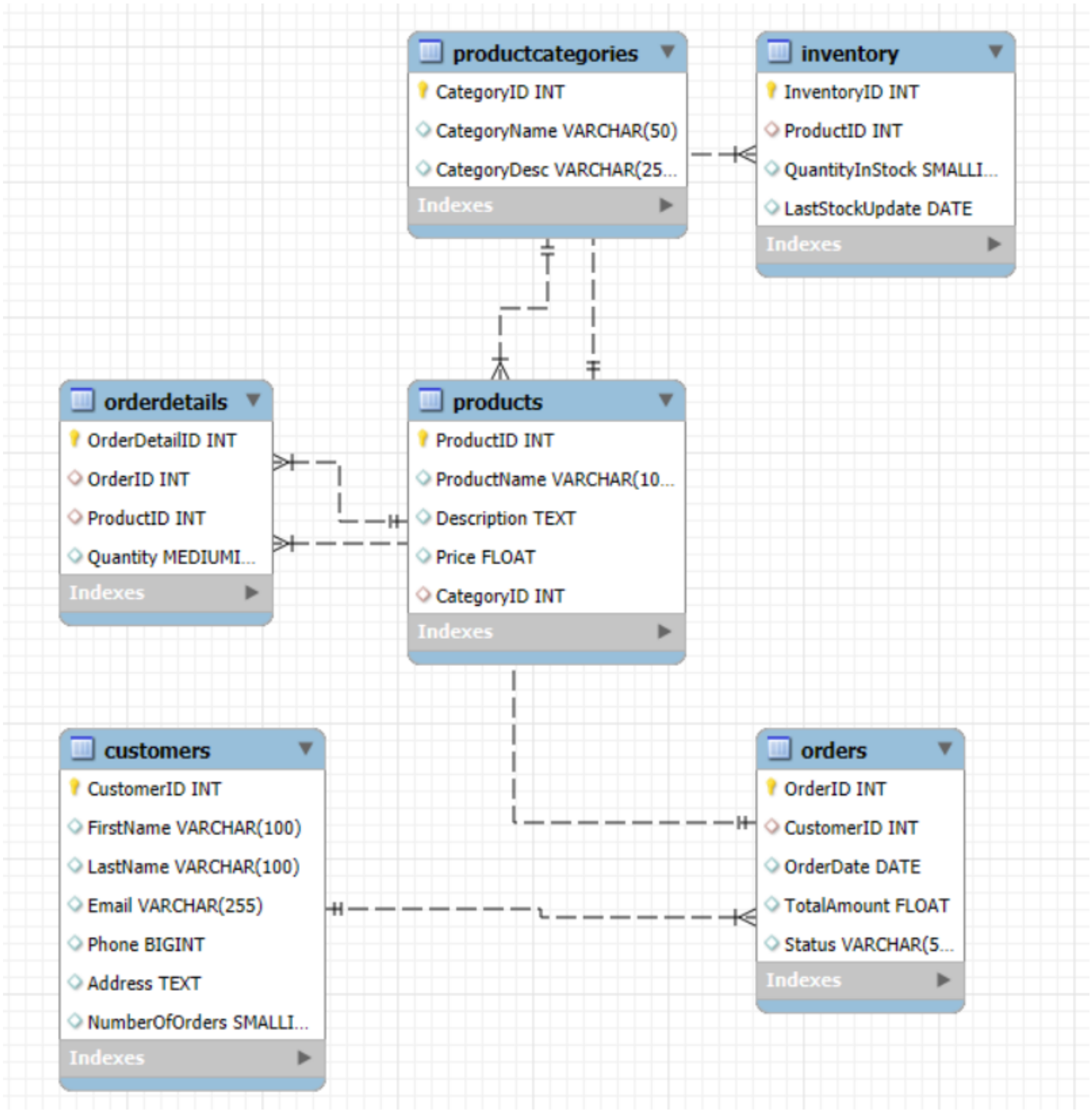


```
mysql> ALTER TABLE Products MODIFY COLUMN Price FLOAT;
Query OK, 0 rows affected (0.12 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Orders MODIFY COLUMN TotalAmount FLOAT;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

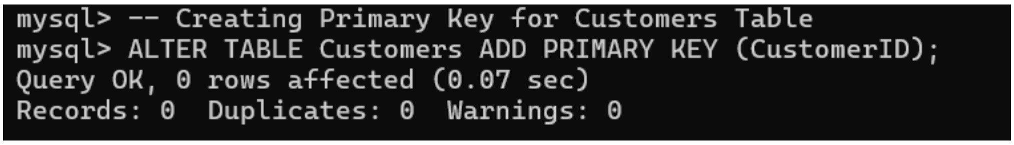


```
mysql> ALTER TABLE Customers MODIFY COLUMN Phone BIGINT;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

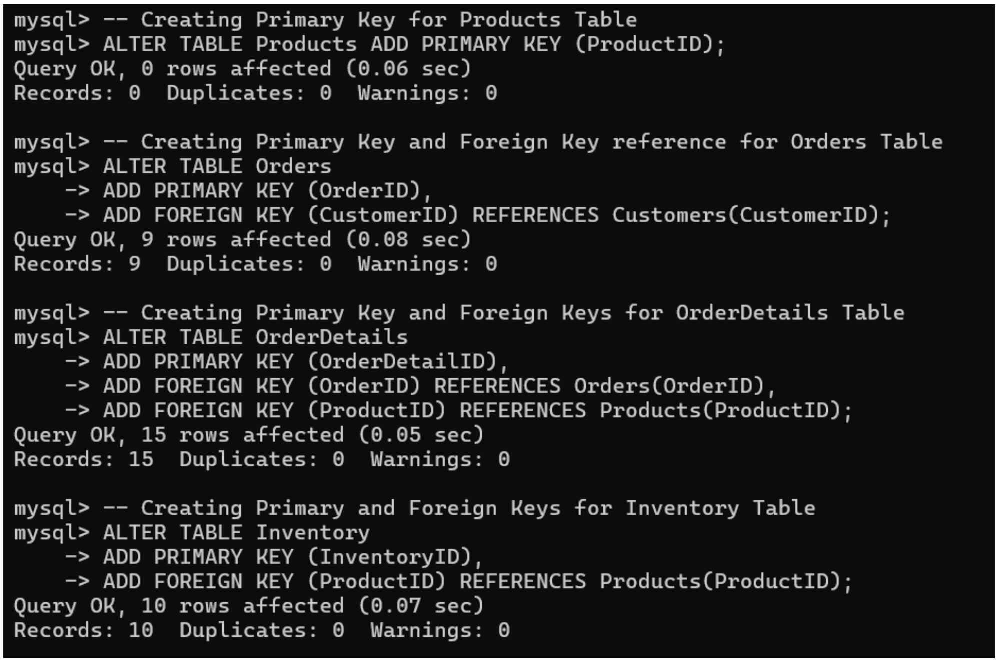
3. Create an ERD (Entity Relationship Diagram) for the database.



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

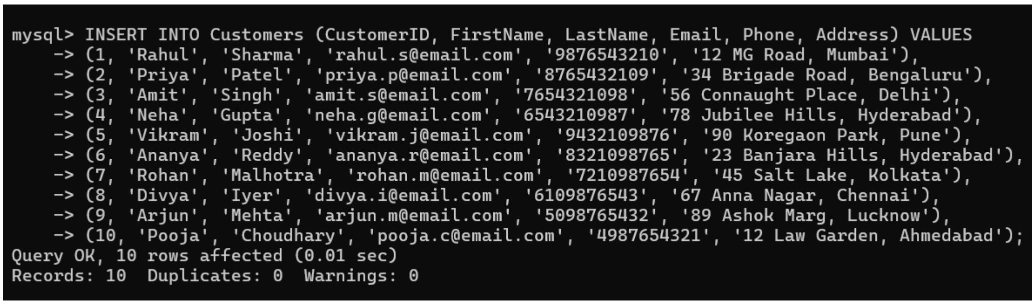


```
mysql> -- Creating Primary Key for Customers Table
mysql> ALTER TABLE Customers ADD PRIMARY KEY (CustomerID);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



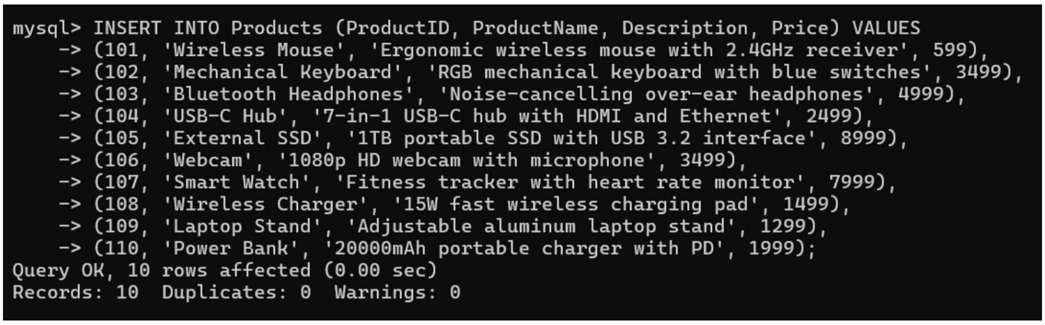
```
mysql> -- Creating Primary Key for Products Table
mysql> ALTER TABLE Products ADD PRIMARY KEY (ProductID);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> -- Creating Primary Key and Foreign Key reference for Orders Table
mysql> ALTER TABLE Orders
    -> ADD PRIMARY KEY (OrderID),
    -> ADD FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID);
Query OK, 9 rows affected (0.08 sec)
Records: 9  Duplicates: 0  Warnings: 0

mysql> -- Creating Primary Key and Foreign Keys for OrderDetails Table
mysql> ALTER TABLE OrderDetails
    -> ADD PRIMARY KEY (OrderDetailID),
    -> ADD FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    -> ADD FOREIGN KEY (ProductID) REFERENCES Products(ProductID);
Query OK, 15 rows affected (0.05 sec)
Records: 15  Duplicates: 0  Warnings: 0

mysql> -- Creating Primary and Foreign Keys for Inventory Table
mysql> ALTER TABLE Inventory
    -> ADD PRIMARY KEY (InventoryID),
    -> ADD FOREIGN KEY (ProductID) REFERENCES Products(ProductID);
Query OK, 10 rows affected (0.07 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

5. Insert at least 10 sample records into each of the following tables.
    a. Customers



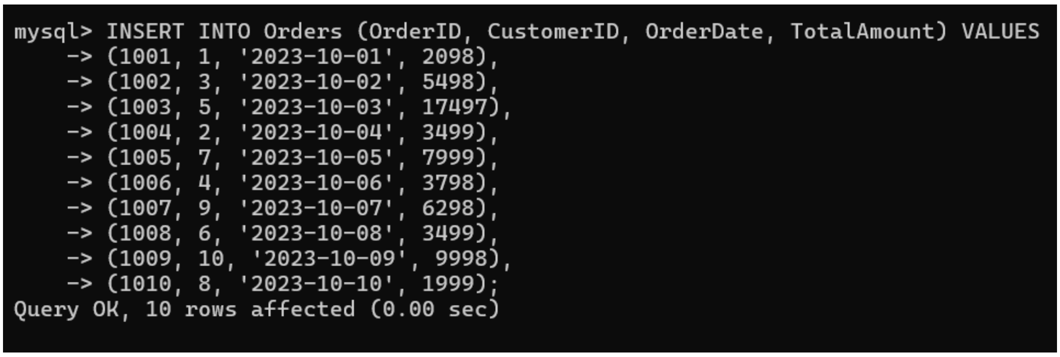
```
mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address) VALUES
    -> (1, 'Rahul', 'Sharma', 'rahul.s@email.com', '9876543210', '12 MG Road, Mumbai'),
    -> (2, 'Priya', 'Patel', 'priya.p@email.com', '8765432109', '34 Brigade Road, Bengaluru'),
    -> (3, 'Amit', 'Singh', 'amit.s@email.com', '7654321098', '56 Connaught Place, Delhi'),
    -> (4, 'Neha', 'Gupta', 'neha.g@email.com', '6543210987', '78 Jubilee Hills, Hyderabad'),
    -> (5, 'Vikram', 'Joshi', 'vikram.j@email.com', '9432109876', '90 Koregaon Park, Pune'),
    -> (6, 'Ananya', 'Reddy', 'ananya.r@email.com', '8321098765', '23 Banjara Hills, Hyderabad'),
    -> (7, 'Rohan', 'Malhotra', 'rohan.m@email.com', '7210987654', '45 Salt Lake, Kolkata'),
    -> (8, 'Divya', 'Iyer', 'divya.i@email.com', '6109876543', '67 Anna Nagar, Chennai'),
    -> (9, 'Arjun', 'Mehta', 'arjun.m@email.com', '5098765432', '89 Ashok Marg, Lucknow'),
    -> (10, 'Pooja', 'Choudhary', 'pooja.c@email.com', '4987654321', '12 Law Garden, Ahmedabad');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

    b. Products



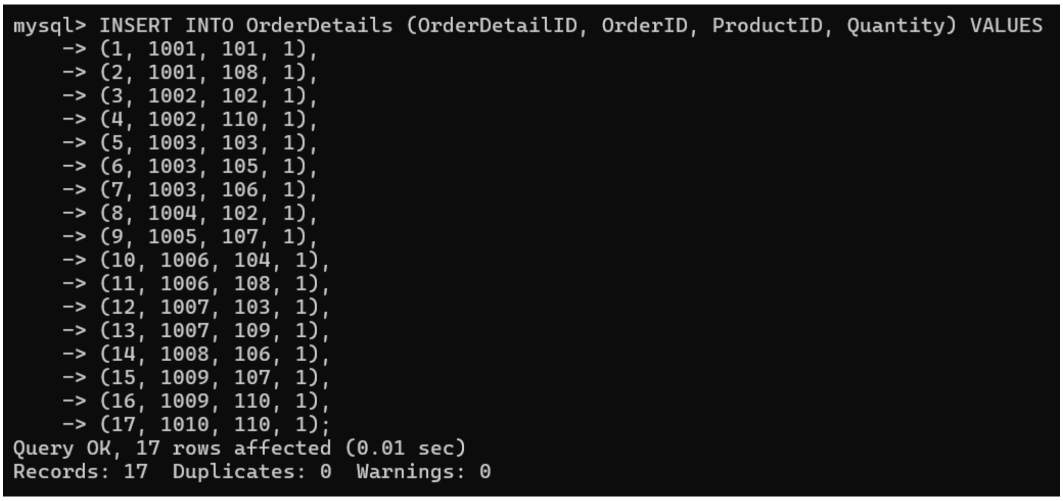
```
mysql> INSERT INTO Products (ProductID, ProductName, Description, Price) VALUES
    -> (101, 'Wireless Mouse', 'Ergonomic wireless mouse with 2.4GHz receiver', 599),
    -> (102, 'Mechanical Keyboard', 'RGB mechanical keyboard with blue switches', 3499),
    -> (103, 'Bluetooth Headphones', 'Noise-cancelling over-ear headphones', 4999),
    -> (104, 'USB-C Hub', '7-in-1 USB-C hub with HDMI and Ethernet', 2499),
    -> (105, 'External SSD', '1TB portable SSD with USB 3.2 interface', 8999),
    -> (106, 'Webcam', '1080p HD webcam with microphone', 3499),
    -> (107, 'Smart Watch', 'Fitness tracker with heart rate monitor', 7999),
    -> (108, 'Wireless Charger', '15W fast wireless charging pad', 1499),
    -> (109, 'Laptop Stand', 'Adjustable aluminum laptop stand', 1299),
    -> (110, 'Power Bank', '20000mAh portable charger with PD', 1999);
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

c. Orders



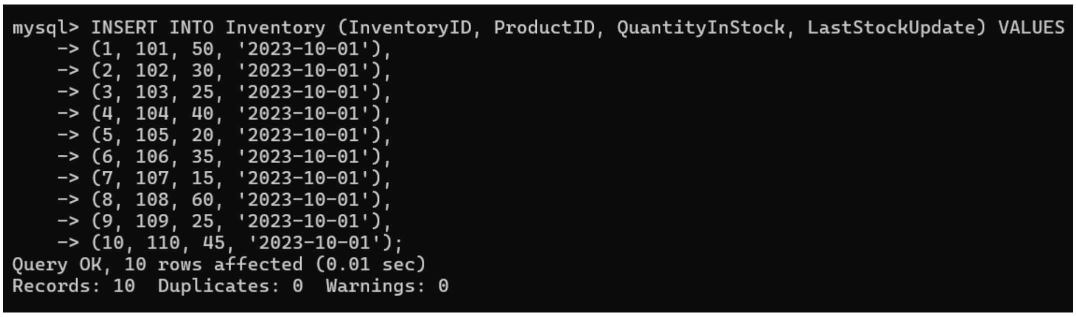
```
mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES
    -> (1001, 1, '2023-10-01', 2098),
    -> (1002, 3, '2023-10-02', 5498),
    -> (1003, 5, '2023-10-03', 17497),
    -> (1004, 2, '2023-10-04', 3499),
    -> (1005, 7, '2023-10-05', 7999),
    -> (1006, 4, '2023-10-06', 3798),
    -> (1007, 9, '2023-10-07', 6298),
    -> (1008, 6, '2023-10-08', 3499),
    -> (1009, 10, '2023-10-09', 9998),
    -> (1010, 8, '2023-10-10', 1999);
Query OK, 10 rows affected (0.00 sec)
```

d. OrderDetails



```
mysql> INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES
    -> (1, 1001, 101, 1),
    -> (2, 1001, 108, 1),
    -> (3, 1002, 102, 1),
    -> (4, 1002, 110, 1),
    -> (5, 1003, 103, 1),
    -> (6, 1003, 105, 1),
    -> (7, 1003, 106, 1),
    -> (8, 1004, 102, 1),
    -> (9, 1005, 107, 1),
    -> (10, 1006, 104, 1),
    -> (11, 1006, 108, 1),
    -> (12, 1007, 103, 1),
    -> (13, 1007, 109, 1),
    -> (14, 1008, 106, 1),
    -> (15, 1009, 107, 1),
    -> (16, 1009, 110, 1),
    -> (17, 1010, 110, 1);
Query OK, 17 rows affected (0.01 sec)
Records: 17  Duplicates: 0  Warnings: 0
```

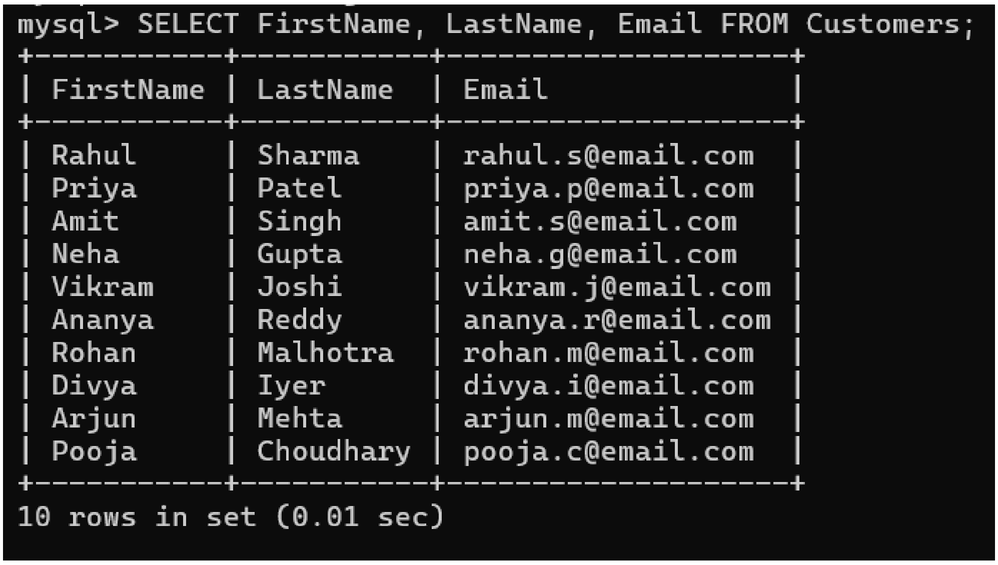
e. Inventory



```
mysql> INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate) VALUES
    -> (1, 101, 50, '2023-10-01'),
    -> (2, 102, 30, '2023-10-01'),
    -> (3, 103, 25, '2023-10-01'),
    -> (4, 104, 40, '2023-10-01'),
    -> (5, 105, 20, '2023-10-01'),
    -> (6, 106, 35, '2023-10-01'),
    -> (7, 107, 15, '2023-10-01'),
    -> (8, 108, 60, '2023-10-01'),
    -> (9, 109, 25, '2023-10-01'),
    -> (10, 110, 45, '2023-10-01');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

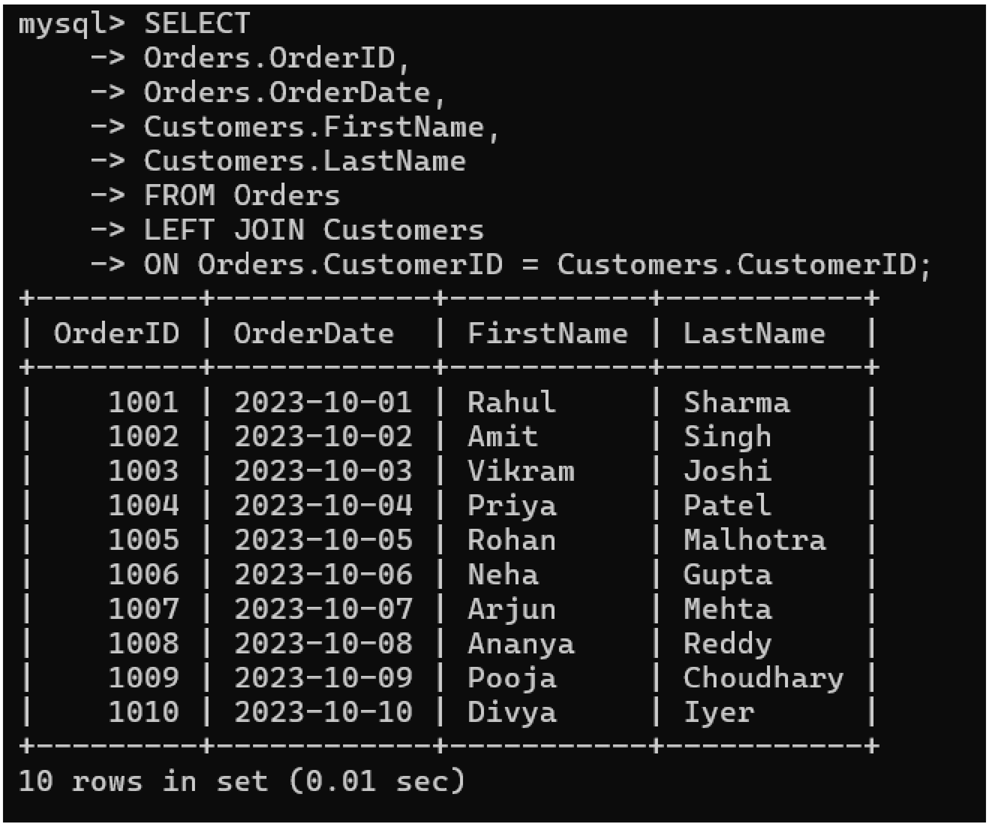
Task 2: Select, Where, Between AND Like

1. Write an SQL query to retrieve the names and emails of all customers.



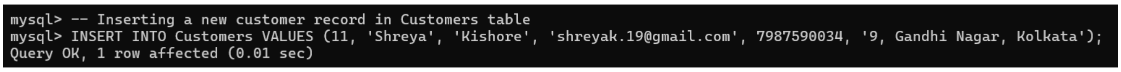
```
mysql> SELECT FirstName, LastName, Email FROM Customers;
+-----------+-----------+--------------------+
| FirstName | LastName  | Email              |
+-----------+-----------+--------------------+
| Rahul     | Sharma    | rahul.s@email.com  |
| Priya     | Patel     | priya.p@email.com  |
| Amit      | Singh     | amit.s@email.com   |
| Neha      | Gupta     | neha.g@email.com   |
| Vikram    | Joshi     | vikram.j@email.com |
| Ananya    | Reddy     | ananya.r@email.com |
| Rohan     | Malhotra  | rohan.m@email.com  |
| Divya     | Iyer      | divya.i@email.com  |
| Arjun     | Mehta     | arjun.m@email.com  |
| Pooja     | Choudhary | pooja.c@email.com  |
+-----------+-----------+--------------------+
10 rows in set (0.01 sec)
```

2. Write an SQL query to list all orders with their order dates and corresponding customer names.

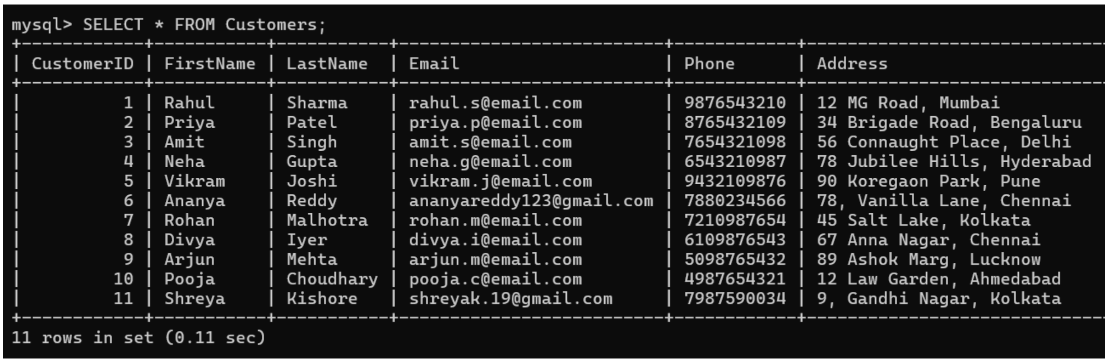


```
mysql> SELECT
    -> Orders.OrderID,
    -> Orders.OrderDate,
    -> Customers.FirstName,
    -> Customers.LastName
    -> FROM Orders
    -> LEFT JOIN Customers
    -> ON Orders.CustomerID = Customers.CustomerID;
+---------+------------+-----------+-----------+
| OrderID | OrderDate  | FirstName | LastName  |
+---------+------------+-----------+-----------+
|    1001 | 2023-10-01 | Rahul     | Sharma    |
|    1002 | 2023-10-02 | Amit      | Singh     |
|    1003 | 2023-10-03 | Vikram    | Joshi     |
|    1004 | 2023-10-04 | Priya     | Patel     |
|    1005 | 2023-10-05 | Rohan     | Malhotra  |
|    1006 | 2023-10-06 | Neha      | Gupta     |
|    1007 | 2023-10-07 | Arjun     | Mehta     |
|    1008 | 2023-10-08 | Ananya    | Reddy     |
|    1009 | 2023-10-09 | Pooja     | Choudhary |
|    1010 | 2023-10-10 | Divya     | Iyer      |
+---------+------------+-----------+-----------+
10 rows in set (0.01 sec)
```

3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.

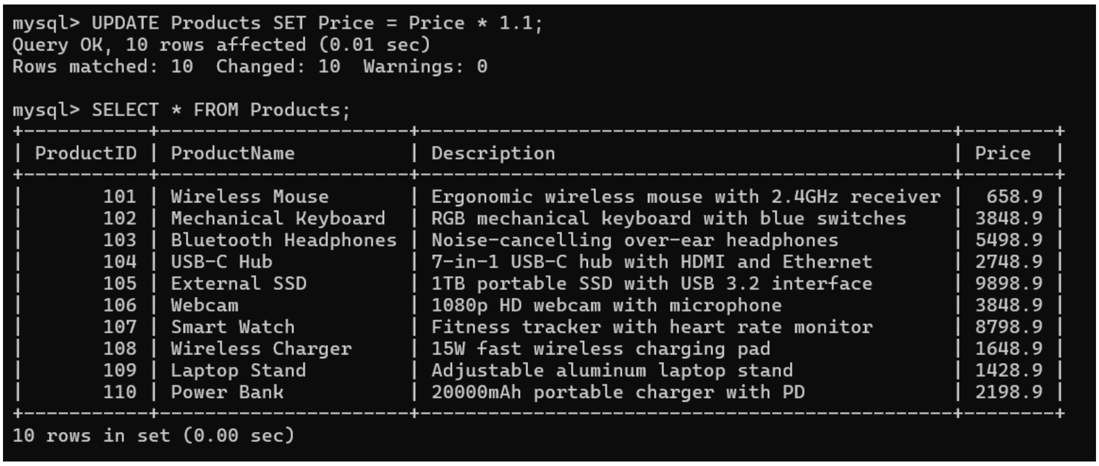


```
mysql> -- Inserting a new customer record in Customers table
mysql> INSERT INTO Customers VALUES (11, 'Shreya', 'Kishore', 'shreyak.19@gmail.com', 7987590034, '9, Gandhi Nagar, Kolkata');
Query OK, 1 row affected (0.01 sec)
```



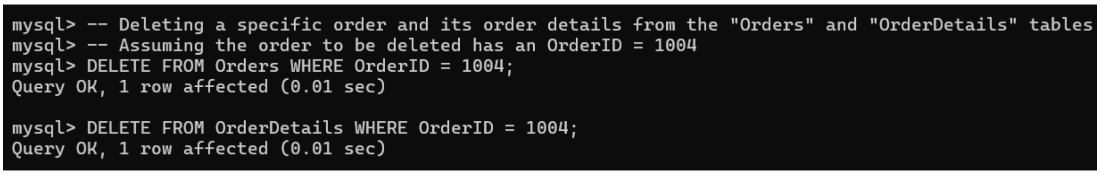
```
mysql> SELECT * FROM Customers;
+------------+-----------+-----------+--------------------------+------------+----------------------------
| CustomerID | FirstName | LastName  | Email                    | Phone      | Address
+------------+-----------+-----------+--------------------------+------------+----------------------------
|          1 | Rahul     | Sharma    | rahul.s@email.com        | 9876543210 | 12 MG Road, Mumbai
|          2 | Priya     | Patel     | priya.p@email.com        | 8765432109 | 34 Brigade Road, Bengaluru
|          3 | Amit      | Singh     | amit.s@email.com         | 7654321098 | 56 Connaught Place, Delhi
|          4 | Neha      | Gupta     | neha.g@email.com         | 6543210987 | 78 Jubilee Hills, Hyderabad
|          5 | Vikram    | Joshi     | vikram.j@email.com       | 9432109876 | 90 Koregaon Park, Pune
|          6 | Ananya    | Reddy     | ananyareddy123@gmail.com | 7880234566 | 78, Vanilla Lane, Chennai
|          7 | Rohan     | Malhotra  | rohan.m@email.com        | 7210987654 | 45 Salt Lake, Kolkata
|          8 | Divya     | Iyer      | divya.i@email.com        | 6109876543 | 67 Anna Nagar, Chennai
|          9 | Arjun     | Mehta     | arjun.m@email.com        | 5098765432 | 89 Ashok Marg, Lucknow
|         10 | Pooja     | Choudhary | pooja.c@email.com        | 4987654321 | 12 Law Garden, Ahmedabad
|         11 | Shreya    | Kishore   | shreyak.19@gmail.com     | 7987590034 | 9, Gandhi Nagar, Kolkata
+------------+-----------+-----------+--------------------------+------------+----------------------------
11 rows in set (0.11 sec)
```

4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.



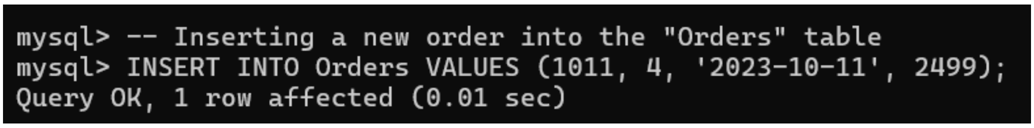
```
mysql> UPDATE Products SET Price = Price * 1.1;
Query OK, 10 rows affected (0.01 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> SELECT * FROM Products;
+-----------+----------------------+-----------------------------------------------+--------+
| ProductID | ProductName          | Description                                   | Price  |
+-----------+----------------------+-----------------------------------------------+--------+
|       101 | Wireless Mouse       | Ergonomic wireless mouse with 2.4GHz receiver |  658.9 |
|       102 | Mechanical Keyboard  | RGB mechanical keyboard with blue switches    | 3848.9 |
|       103 | Bluetooth Headphones | Noise-cancelling over-ear headphones          | 5498.9 |
|       104 | USB-C Hub            | 7-in-1 USB-C hub with HDMI and Ethernet       | 2748.9 |
|       105 | External SSD         | 1TB portable SSD with USB 3.2 interface       | 9898.9 |
|       106 | Webcam               | 1080p HD webcam with microphone               | 3848.9 |
|       107 | Smart Watch          | Fitness tracker with heart rate monitor       | 8798.9 |
|       108 | Wireless Charger     | 15W fast wireless charging pad                | 1648.9 |
|       109 | Laptop Stand         | Adjustable aluminum laptop stand              | 1428.9 |
|       110 | Power Bank           | 20000mAh portable charger with PD             | 2198.9 |
+-----------+----------------------+-----------------------------------------------+--------+
10 rows in set (0.00 sec)
```

5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.



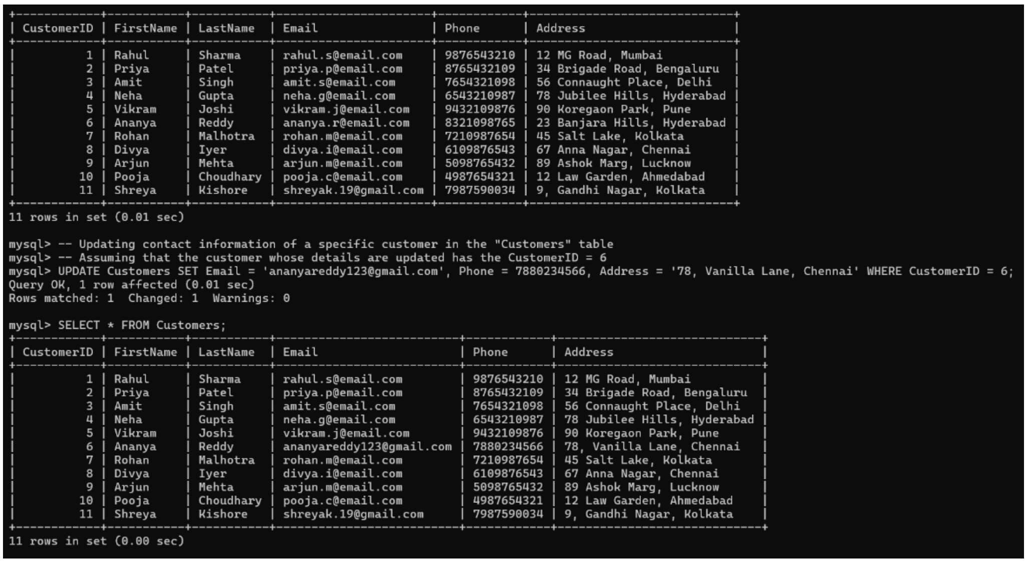
```
mysql> -- Deleting a specific order and its order details from the "Orders" and "OrderDetails" tables
mysql> -- Assuming the order to be deleted has an OrderID = 1004
mysql> DELETE FROM Orders WHERE OrderID = 1004;
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM OrderDetails WHERE OrderID = 1004;
Query OK, 1 row affected (0.01 sec)
```

6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.



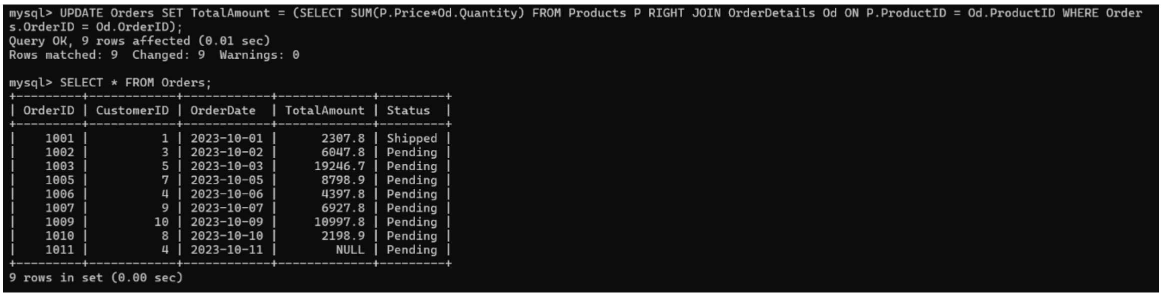
```
mysql> -- Inserting a new order into the "Orders" table
mysql> INSERT INTO Orders VALUES (1011, 4, '2023-10-11', 2499);
Query OK, 1 row affected (0.01 sec)
```

7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.



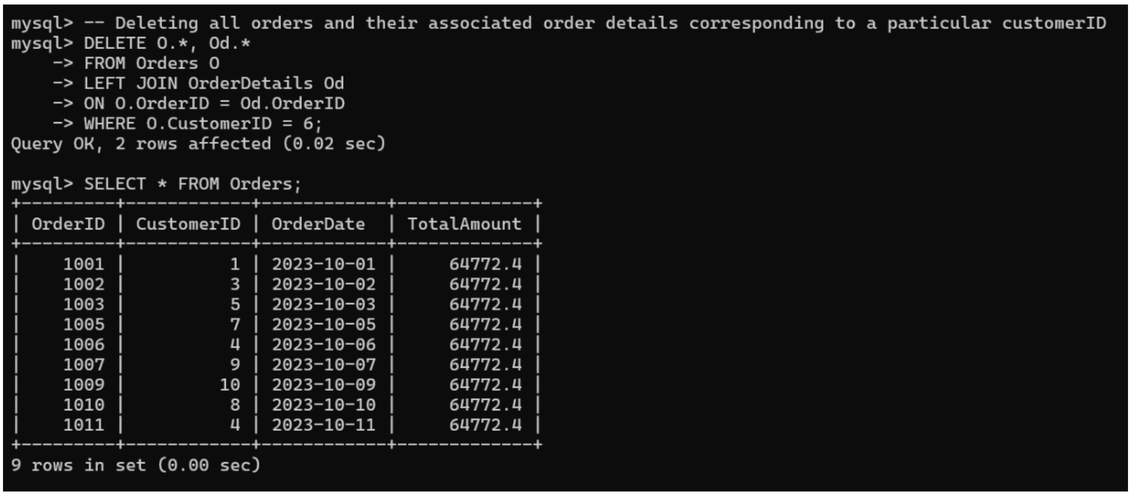
```
+------------+-----------+-----------+----------------------+------------+-----------------------------+
| CustomerID | FirstName | LastName  | Email                | Phone      | Address                     |
+------------+-----------+-----------+----------------------+------------+-----------------------------+
|          1 | Rahul     | Sharma    | rahul.s@email.com    | 9876543210 | 12 MG Road, Mumbai          |
|          2 | Priya     | Patel     | priya.p@email.com    | 8765432109 | 34 Brigade Road, Bengaluru  |
|          3 | Amit      | Singh     | amit.s@email.com     | 7654321098 | 56 Connaught Place, Delhi   |
|          4 | Neha      | Gupta     | neha.g@email.com     | 6543210987 | 78 Jubilee Hills, Hyderabad |
|          5 | Vikram    | Joshi     | vikram.j@email.com   | 9432109876 | 90 Koregaon Park, Pune      |
|          6 | Ananya    | Reddy     | ananya.r@email.com   | 8321098765 | 23 Banjara Hills, Hyderabad |
|          7 | Rohan     | Malhotra  | rohan.m@email.com    | 7210987654 | 45 Salt Lake, Kolkata       |
|          8 | Divya     | Iyer      | divya.i@email.com    | 6109876543 | 67 Anna Nagar, Chennai      |
|          9 | Arjun     | Mehta     | arjun.m@email.com    | 5098765432 | 89 Ashok Marg, Lucknow      |
|         10 | Pooja     | Choudhary | pooja.c@email.com    | 4987654321 | 12 Law Garden, Ahmedabad    |
|         11 | Shreya    | Kishore   | shreyak.19@gmail.com | 7987590034 | 9, Gandhi Nagar, Kolkata    |
+------------+-----------+-----------+----------------------+------------+-----------------------------+
11 rows in set (0.01 sec)

mysql> -- Updating contact information of a specific customer in the "Customers" table
mysql> -- Assuming that the customer whose details are updated has the CustomerID = 6
mysql> UPDATE Customers SET Email = 'ananyareddy123@gmail.com', Phone = 7880234566, Address = '78, Vanilla Lane, Chennai' WHERE CustomerID = 6;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Customers;
+------------+-----------+-----------+--------------------------+------------+-----------------------------+
| CustomerID | FirstName | LastName  | Email                    | Phone      | Address                     |
+------------+-----------+-----------+--------------------------+------------+-----------------------------+
|          1 | Rahul     | Sharma    | rahul.s@email.com        | 9876543210 | 12 MG Road, Mumbai          |
|          2 | Priya     | Patel     | priya.p@email.com        | 8765432109 | 34 Brigade Road, Bengaluru  |
|          3 | Amit      | Singh     | amit.s@email.com         | 7654321098 | 56 Connaught Place, Delhi   |
|          4 | Neha      | Gupta     | neha.g@email.com         | 6543210987 | 78 Jubilee Hills, Hyderabad |
|          5 | Vikram    | Joshi     | vikram.j@email.com       | 9432109876 | 90 Koregaon Park, Pune      |
|          6 | Ananya    | Reddy     | ananyareddy123@gmail.com | 7880234566 | 78, Vanilla Lane, Chennai   |
|          7 | Rohan     | Malhotra  | rohan.m@email.com        | 7210987654 | 45 Salt Lake, Kolkata       |
|          8 | Divya     | Iyer      | divya.i@email.com        | 6109876543 | 67 Anna Nagar, Chennai      |
|          9 | Arjun     | Mehta     | arjun.m@email.com        | 5098765432 | 89 Ashok Marg, Lucknow      |
|         10 | Pooja     | Choudhary | pooja.c@email.com        | 4987654321 | 12 Law Garden, Ahmedabad    |
|         11 | Shreya    | Kishore   | shreyak.19@gmail.com     | 7987590034 | 9, Gandhi Nagar, Kolkata    |
+------------+-----------+-----------+--------------------------+------------+-----------------------------+
11 rows in set (0.00 sec)
```

8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.

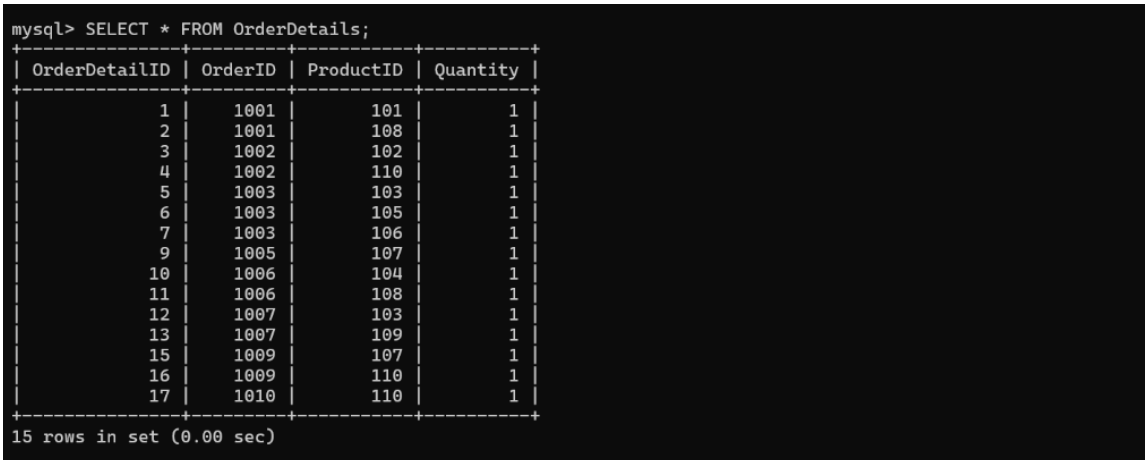


```
mysql> UPDATE Orders SET TotalAmount = (SELECT SUM(P.Price*Od.Quantity) FROM Products P RIGHT JOIN OrderDetails Od ON P.ProductID = Od.ProductID WHERE Order
s.OrderID = Od.OrderID);
Query OK, 9 rows affected (0.01 sec)
Rows matched: 9  Changed: 9  Warnings: 0

mysql> SELECT * FROM Orders;
+---------+------------+------------+-------------+---------+
| OrderID | CustomerID | OrderDate  | TotalAmount | Status  |
+---------+------------+------------+-------------+---------+
|    1001 |          1 | 2023-10-01 |      2307.8 | Shipped |
|    1002 |          3 | 2023-10-02 |      6047.8 | Pending |
|    1003 |          5 | 2023-10-03 |     19246.7 | Pending |
|    1005 |          7 | 2023-10-05 |      8798.9 | Pending |
|    1006 |          4 | 2023-10-06 |      4397.8 | Pending |
|    1007 |          9 | 2023-10-07 |      6927.8 | Pending |
|    1009 |         10 | 2023-10-09 |     10997.8 | Pending |
|    1010 |          8 | 2023-10-10 |      2198.9 | Pending |
|    1011 |          4 | 2023-10-11 |        NULL | Pending |
+---------+------------+------------+-------------+---------+
9 rows in set (0.00 sec)
```

9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.

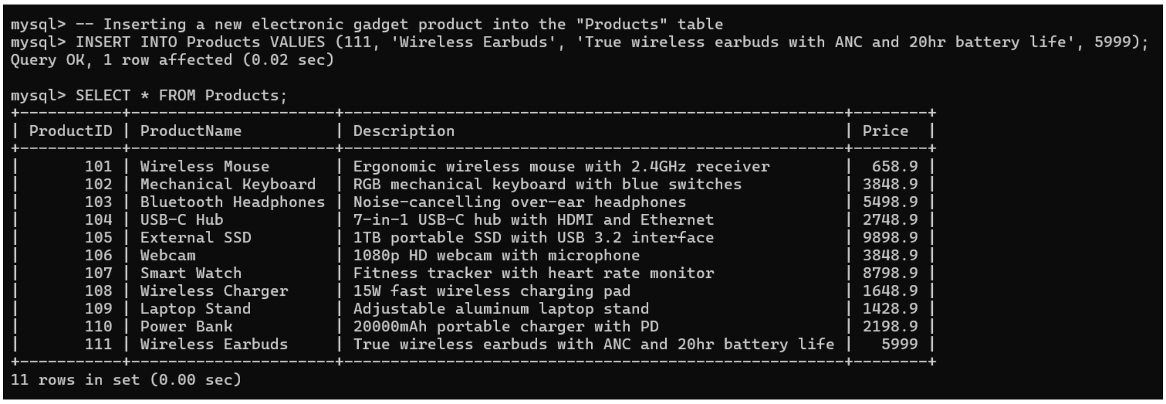


```
mysql> -- Deleting all orders and their associated order details corresponding to a particular customerID
mysql> DELETE O.*, Od.*
    -> FROM Orders O
    -> LEFT JOIN OrderDetails Od
    -> ON O.OrderID = Od.OrderID
    -> WHERE O.CustomerID = 6;
Query OK, 2 rows affected (0.02 sec)

mysql> SELECT * FROM Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|    1001 |          1 | 2023-10-01 |     64772.4 |
|    1002 |          3 | 2023-10-02 |     64772.4 |
|    1003 |          5 | 2023-10-03 |     64772.4 |
|    1005 |          7 | 2023-10-05 |     64772.4 |
|    1006 |          4 | 2023-10-06 |     64772.4 |
|    1007 |          9 | 2023-10-07 |     64772.4 |
|    1009 |         10 | 2023-10-09 |     64772.4 |
|    1010 |          8 | 2023-10-10 |     64772.4 |
|    1011 |          4 | 2023-10-11 |     64772.4 |
+---------+------------+------------+-------------+
9 rows in set (0.00 sec)
```



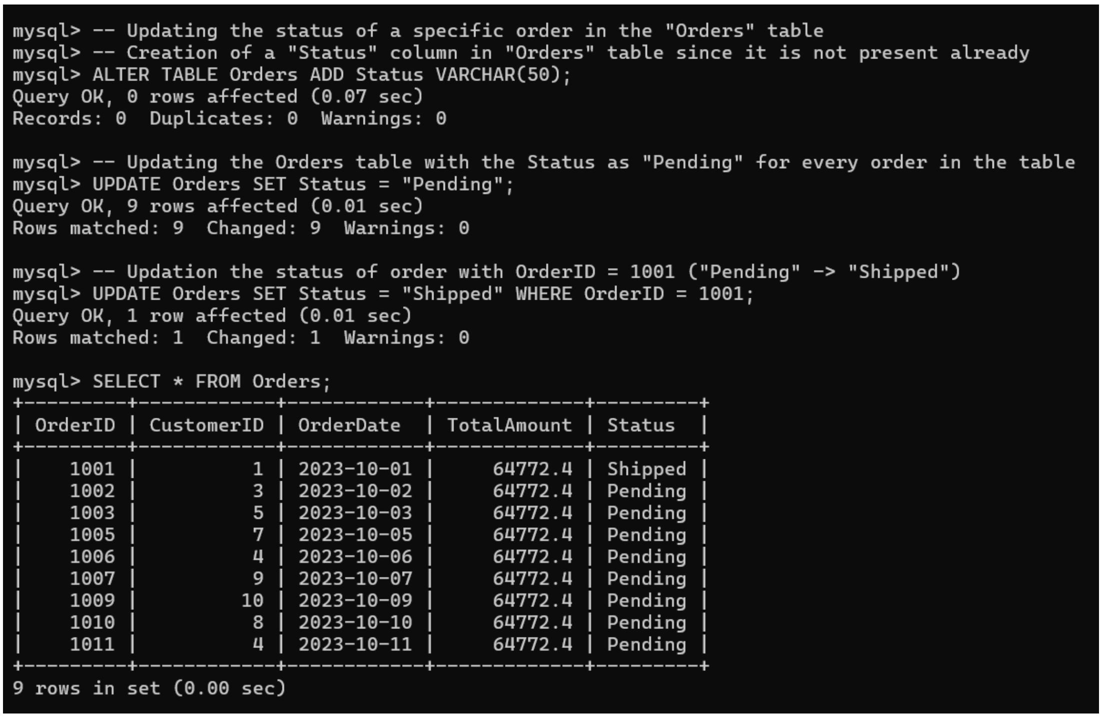
```
mysql> SELECT * FROM OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|             1 |    1001 |       101 |        1 |
|             2 |    1001 |       108 |        1 |
|             3 |    1002 |       102 |        1 |
|             4 |    1002 |       110 |        1 |
|             5 |    1003 |       103 |        1 |
|             6 |    1003 |       105 |        1 |
|             7 |    1003 |       106 |        1 |
|             9 |    1005 |       107 |        1 |
|            10 |    1006 |       104 |        1 |
|            11 |    1006 |       108 |        1 |
|            12 |    1007 |       103 |        1 |
|            13 |    1007 |       109 |        1 |
|            15 |    1009 |       107 |        1 |
|            16 |    1009 |       110 |        1 |
|            17 |    1010 |       110 |        1 |
+---------------+---------+-----------+----------+
15 rows in set (0.00 sec)
```

10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.



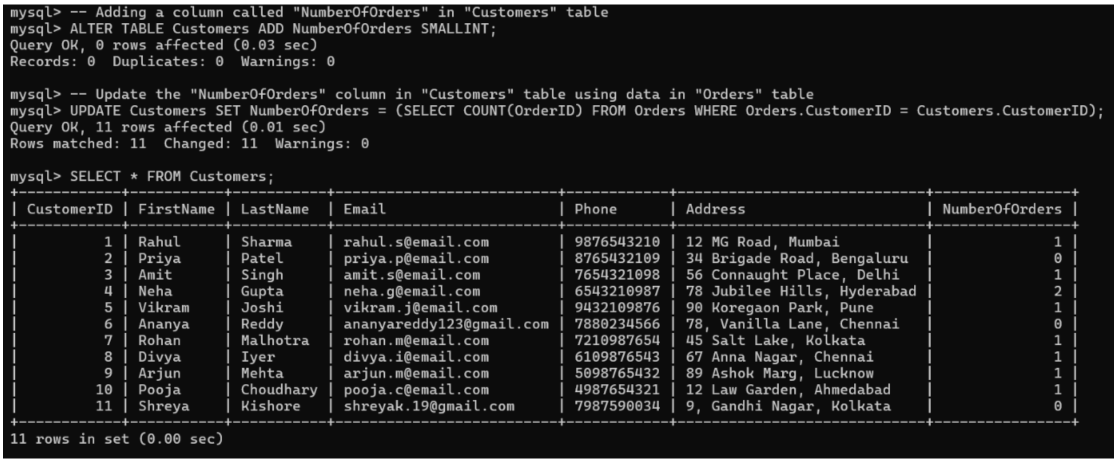
```
mysql> -- Inserting a new electronic gadget product into the "Products" table
mysql> INSERT INTO Products VALUES (111, 'Wireless Earbuds', 'True wireless earbuds with ANC and 20hr battery life', 5999);
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM Products;
+-----------+----------------------+------------------------------------------------------+--------+
| ProductID | ProductName          | Description                                          | Price  |
+-----------+----------------------+------------------------------------------------------+--------+
|       101 | Wireless Mouse       | Ergonomic wireless mouse with 2.4GHz receiver        |  658.9 |
|       102 | Mechanical Keyboard  | RGB mechanical keyboard with blue switches           | 3848.9 |
|       103 | Bluetooth Headphones | Noise-cancelling over-ear headphones                 | 5498.9 |
|       104 | USB-C Hub            | 7-in-1 USB-C hub with HDMI and Ethernet              | 2748.9 |
|       105 | External SSD         | 1TB portable SSD with USB 3.2 interface              | 9898.9 |
|       106 | Webcam               | 1080p HD webcam with microphone                      | 3848.9 |
|       107 | Smart Watch          | Fitness tracker with heart rate monitor              | 8798.9 |
|       108 | Wireless Charger     | 15W fast wireless charging pad                       | 1648.9 |
|       109 | Laptop Stand         | Adjustable aluminum laptop stand                     | 1428.9 |
|       110 | Power Bank           | 20000mAh portable charger with PD                    | 2198.9 |
|       111 | Wireless Earbuds     | True wireless earbuds with ANC and 20hr battery life |   5999 |
+-----------+----------------------+------------------------------------------------------+--------+
11 rows in set (0.00 sec)
```

11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.



```
mysql> -- Updating the status of a specific order in the "Orders" table
mysql> -- Creation of a "Status" column in "Orders" table since it is not present already
mysql> ALTER TABLE Orders ADD Status VARCHAR(50);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> -- Updating the Orders table with the Status as "Pending" for every order in the table
mysql> UPDATE Orders SET Status = "Pending";
Query OK, 9 rows affected (0.01 sec)
Rows matched: 9  Changed: 9  Warnings: 0

mysql> -- Updation the status of order with OrderID = 1001 ("Pending" -> "Shipped")
mysql> UPDATE Orders SET Status = "Shipped" WHERE OrderID = 1001;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM Orders;
+---------+------------+------------+-------------+---------+
| OrderID | CustomerID | OrderDate  | TotalAmount | Status  |
+---------+------------+------------+-------------+---------+
|    1001 |          1 | 2023-10-01 |     64772.4 | Shipped |
|    1002 |          3 | 2023-10-02 |     64772.4 | Pending |
|    1003 |          5 | 2023-10-03 |     64772.4 | Pending |
|    1005 |          7 | 2023-10-05 |     64772.4 | Pending |
|    1006 |          4 | 2023-10-06 |     64772.4 | Pending |
|    1007 |          9 | 2023-10-07 |     64772.4 | Pending |
|    1009 |         10 | 2023-10-09 |     64772.4 | Pending |
|    1010 |          8 | 2023-10-10 |     64772.4 | Pending |
|    1011 |          4 | 2023-10-11 |     64772.4 | Pending |
+---------+------------+------------+-------------+---------+
9 rows in set (0.00 sec)
```

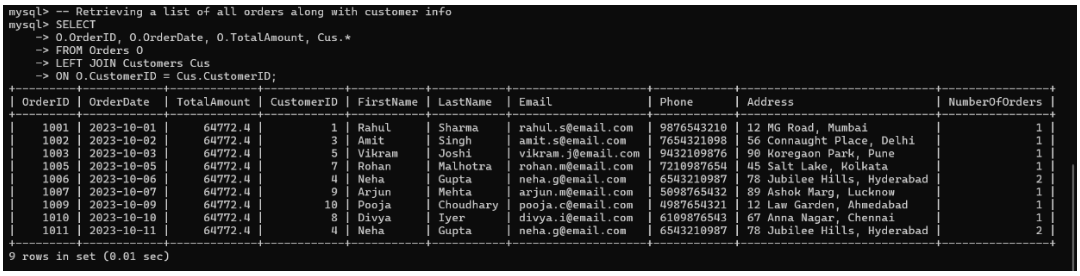
12. Write an SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table.



```
mysql> -- Adding a column called "NumberOfOrders" in "Customers" table
mysql> ALTER TABLE Customers ADD NumberOfOrders SMALLINT;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> -- Update the "NumberOfOrders" column in "Customers" table using data in "Orders" table
mysql> UPDATE Customers SET NumberOfOrders = (SELECT COUNT(OrderID) FROM Orders WHERE Orders.CustomerID = Customers.CustomerID);
Query OK, 11 rows affected (0.01 sec)
Rows matched: 11  Changed: 11  Warnings: 0

mysql> SELECT * FROM Customers;
+------------+-----------+-----------+--------------------------+------------+-----------------------------+----------------+
| CustomerID | FirstName | LastName  | Email                    | Phone      | Address                     | NumberOfOrders |
+------------+-----------+-----------+--------------------------+------------+-----------------------------+----------------+
|          1 | Rahul     | Sharma    | rahul.s@email.com        | 9876543210 | 12 MG Road, Mumbai          |              1 |
|          2 | Priya     | Patel     | priya.p@email.com        | 8765432109 | 34 Brigade Road, Bengaluru  |              0 |
|          3 | Amit      | Singh     | amit.s@email.com         | 7654321098 | 56 Connaught Place, Delhi   |              1 |
|          4 | Neha      | Gupta     | neha.g@email.com         | 6543210987 | 78 Jubilee Hills, Hyderabad |              2 |
|          5 | Vikram    | Joshi     | vikram.j@email.com       | 9432109876 | 90 Koregaon Park, Pune      |              1 |
|          6 | Ananya    | Reddy     | ananyareddy123@gmail.com | 7880234566 | 78, Vanilla Lane, Chennai   |              0 |
|          7 | Rohan     | Malhotra  | rohan.m@email.com        | 7210987654 | 45 Salt Lake, Kolkata       |              1 |
|          8 | Divya     | Iyer      | divya.i@email.com        | 6109876543 | 67 Anna Nagar, Chennai      |              1 |
|          9 | Arjun     | Mehta     | arjun.m@email.com        | 5098765432 | 89 Ashok Marg, Lucknow      |              1 |
|         10 | Pooja     | Choudhary | pooja.c@email.com        | 4987654321 | 12 Law Garden, Ahmedabad    |              1 |
|         11 | Shreya    | Kishore   | shreyak.19@gmail.com     | 7987590034 | 9, Gandhi Nagar, Kolkata    |              0 |
+------------+-----------+-----------+--------------------------+------------+-----------------------------+----------------+
11 rows in set (0.00 sec)
```

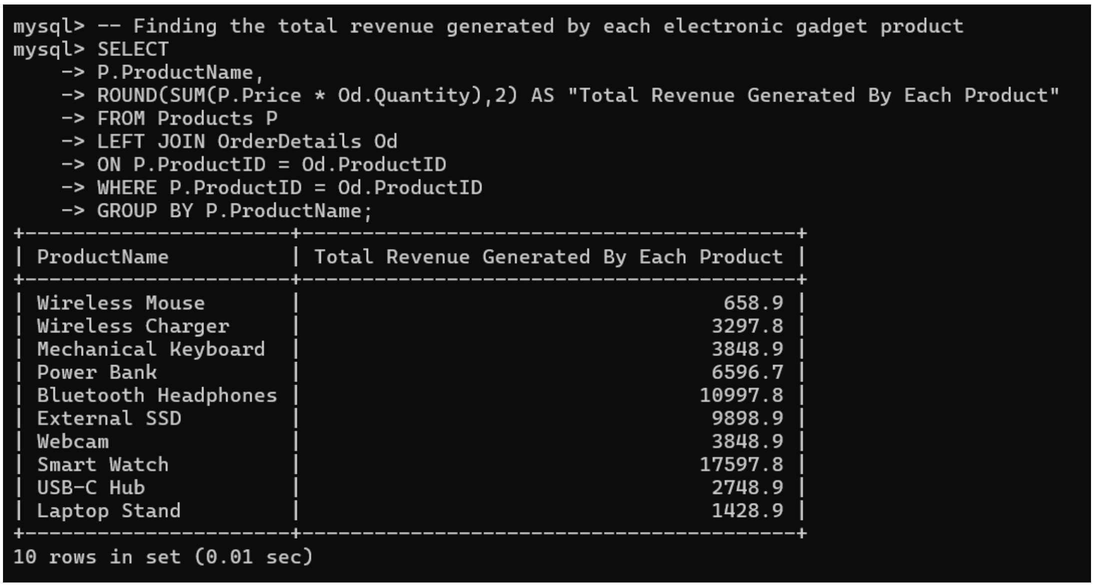
Task 3: Aggregate functions, Having, Order By, Group By and Joins

1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.



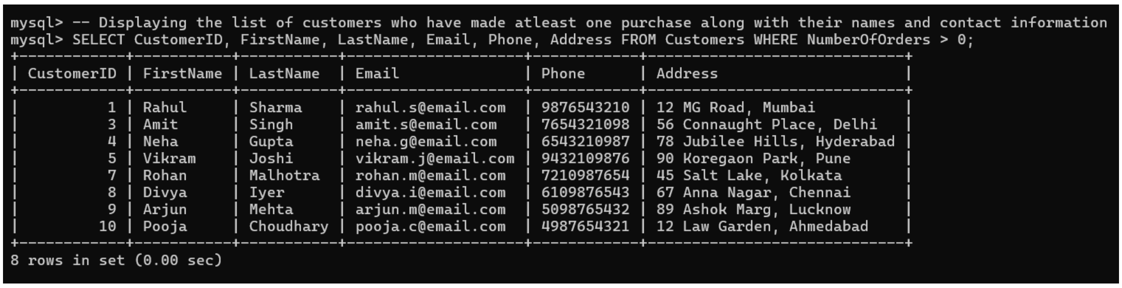
```
mysql> -- Retrieving a list of all orders along with customer info
mysql> SELECT
    -> O.OrderID, O.OrderDate, O.TotalAmount, Cus.*
    -> FROM Orders O
    -> LEFT JOIN Customers Cus
    -> ON O.CustomerID = Cus.CustomerID;
+---------+------------+-------------+------------+-----------+-----------+--------------------+------------+-----------------------------+----------------+
| OrderID | OrderDate  | TotalAmount | CustomerID | FirstName | LastName  | Email              | Phone      | Address                     | NumberOfOrders |
+---------+------------+-------------+------------+-----------+-----------+--------------------+------------+-----------------------------+----------------+
|    1001 | 2023-10-01 |     64772.4 |          1 | Rahul     | Sharma    | rahul.s@email.com  | 9876543210 | 12 MG Road, Mumbai          |              1 |
|    1002 | 2023-10-02 |     64772.4 |          3 | Amit      | Singh     | amit.s@email.com   | 7654321098 | 56 Connaught Place, Delhi   |              1 |
|    1003 | 2023-10-03 |     64772.4 |          5 | Vikram    | Joshi     | vikram.j@email.com | 9432109876 | 90 Koregaon Park, Pune      |              1 |
|    1005 | 2023-10-05 |     64772.4 |          7 | Rohan     | Malhotra  | rohan.m@email.com  | 7210987654 | 45 Salt Lake, Kolkata       |              1 |
|    1006 | 2023-10-06 |     64772.4 |          4 | Neha      | Gupta     | neha.g@email.com   | 6543210987 | 78 Jubilee Hills, Hyderabad |              2 |
|    1007 | 2023-10-07 |     64772.4 |          9 | Arjun     | Mehta     | arjun.m@email.com  | 5098765432 | 89 Ashok Marg, Lucknow      |              1 |
|    1009 | 2023-10-09 |     64772.4 |         10 | Pooja     | Choudhary | pooja.c@email.com  | 4987654321 | 12 Law Garden, Ahmedabad    |              1 |
|    1010 | 2023-10-10 |     64772.4 |          8 | Divya     | Iyer      | divya.i@email.com  | 6109876543 | 67 Anna Nagar, Chennai      |              1 |
|    1011 | 2023-10-11 |     64772.4 |          4 | Neha      | Gupta     | neha.g@email.com   | 6543210987 | 78 Jubilee Hills, Hyderabad |              2 |
+---------+------------+-------------+------------+-----------+-----------+--------------------+------------+-----------------------------+----------------+
9 rows in set (0.01 sec)
```

2. Write an SQL query to find the total revenue generated by each electronic gadget product. Include the product name and the total revenue.



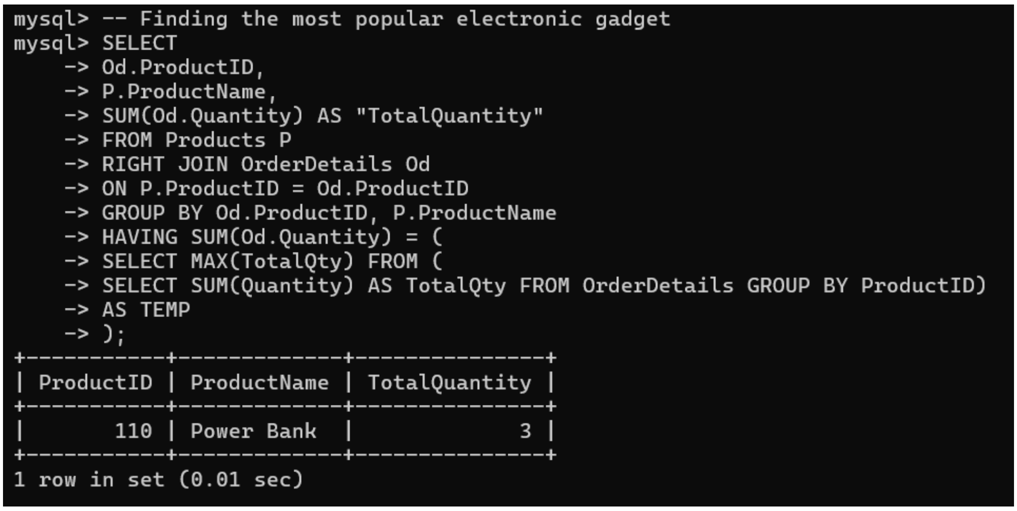
```
mysql> -- Finding the total revenue generated by each electronic gadget product
mysql> SELECT
    -> P.ProductName,
    -> ROUND(SUM(P.Price * Od.Quantity),2) AS "Total Revenue Generated By Each Product"
    -> FROM Products P
    -> LEFT JOIN OrderDetails Od
    -> ON P.ProductID = Od.ProductID
    -> WHERE P.ProductID = Od.ProductID
    -> GROUP BY P.ProductName;
+----------------------+-----------------------------------------+
| ProductName          | Total Revenue Generated By Each Product |
+----------------------+-----------------------------------------+
| Wireless Mouse       |                                   658.9 |
| Wireless Charger     |                                  3297.8 |
| Mechanical Keyboard  |                                  3848.9 |
| Power Bank           |                                  6596.7 |
| Bluetooth Headphones |                                 10997.8 |
| External SSD         |                                  9898.9 |
| Webcam               |                                  3848.9 |
| Smart Watch          |                                 17597.8 |
| USB-C Hub            |                                  2748.9 |
| Laptop Stand         |                                  1428.9 |
+----------------------+-----------------------------------------+
10 rows in set (0.01 sec)
```

3. Write an SQL query to list all customers who have made at least one purchase. Include their names and contact information.



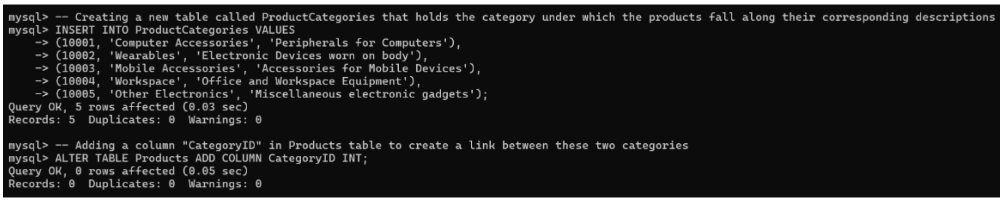
```
mysql> -- Displaying the list of customers who have made atleast one purchase along with their names and contact information
mysql> SELECT CustomerID, FirstName, LastName, Email, Phone, Address FROM Customers WHERE NumberOfOrders > 0;
+------------+-----------+-----------+--------------------+------------+-----------------------------+
| CustomerID | FirstName | LastName  | Email              | Phone      | Address                     |
+------------+-----------+-----------+--------------------+------------+-----------------------------+
|          1 | Rahul     | Sharma    | rahul.s@email.com  | 9876543210 | 12 MG Road, Mumbai          |
|          3 | Amit      | Singh     | amit.s@email.com   | 7654321098 | 56 Connaught Place, Delhi   |
|          4 | Neha      | Gupta     | neha.g@email.com   | 6543210987 | 78 Jubilee Hills, Hyderabad |
|          5 | Vikram    | Joshi     | vikram.j@email.com | 9432109876 | 90 Koregaon Park, Pune      |
|          7 | Rohan     | Malhotra  | rohan.m@email.com  | 7210987654 | 45 Salt Lake, Kolkata       |
|          8 | Divya     | Iyer      | divya.i@email.com  | 6109876543 | 67 Anna Nagar, Chennai      |
|          9 | Arjun     | Mehta     | arjun.m@email.com  | 5098765432 | 89 Ashok Marg, Lucknow      |
|         10 | Pooja     | Choudhary | pooja.c@email.com  | 4987654321 | 12 Law Garden, Ahmedabad    |
+------------+-----------+-----------+--------------------+------------+-----------------------------+
8 rows in set (0.00 sec)
```

4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.

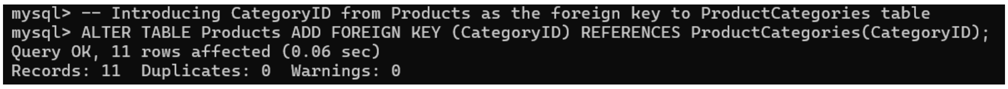


```
mysql> -- Finding the most popular electronic gadget
mysql> SELECT
    -> Od.ProductID,
    -> P.ProductName,
    -> SUM(Od.Quantity) AS "TotalQuantity"
    -> FROM Products P
    -> RIGHT JOIN OrderDetails Od
    -> ON P.ProductID = Od.ProductID
    -> GROUP BY Od.ProductID, P.ProductName
    -> HAVING SUM(Od.Quantity) = (
    -> SELECT MAX(TotalQty) FROM (
    -> SELECT SUM(Quantity) AS TotalQty FROM OrderDetails GROUP BY ProductID)
    -> AS TEMP
    -> );
+-----------+-------------+---------------+
| ProductID | ProductName | TotalQuantity |
+-----------+-------------+---------------+
|       110 | Power Bank  |             3 |
+-----------+-------------+---------------+
1 row in set (0.01 sec)
```

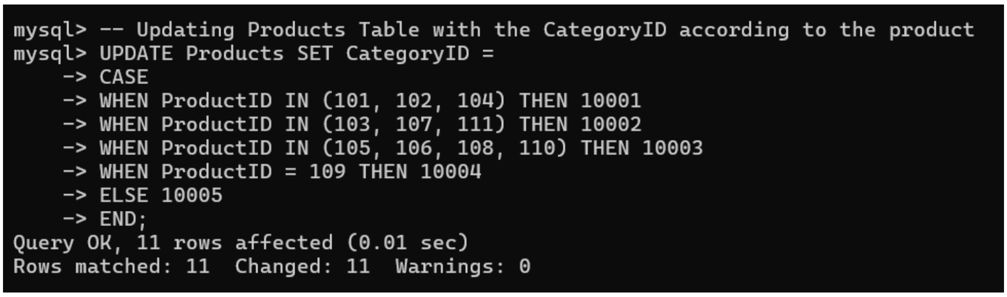
5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.



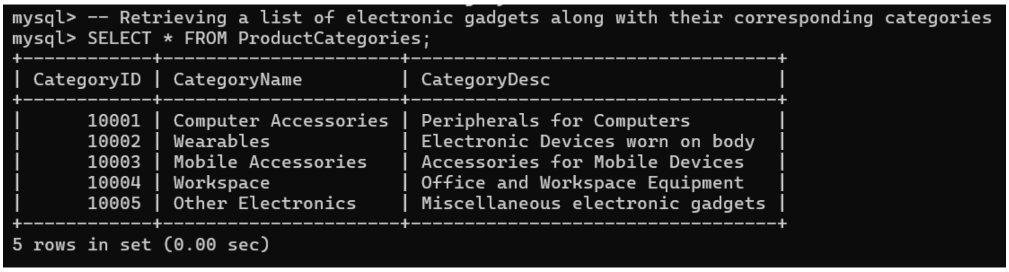
```
mysql> -- Creating a new table called ProductCategories that holds the category under which the products fall along their corresponding descriptions
mysql> INSERT INTO ProductCategories VALUES
    -> (10001, 'Computer Accessories', 'Peripherals for Computers'),
    -> (10002, 'Wearables', 'Electronic Devices worn on body'),
    -> (10003, 'Mobile Accessories', 'Accessories for Mobile Devices'),
    -> (10004, 'Workspace', 'Office and Workspace Equipment'),
    -> (10005, 'Other Electronics', 'Miscellaneous electronic gadgets');
Query OK, 5 rows affected (0.03 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> -- Adding a column "CategoryID" in Products table to create a link between these two categories
mysql> ALTER TABLE Products ADD COLUMN CategoryID INT;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



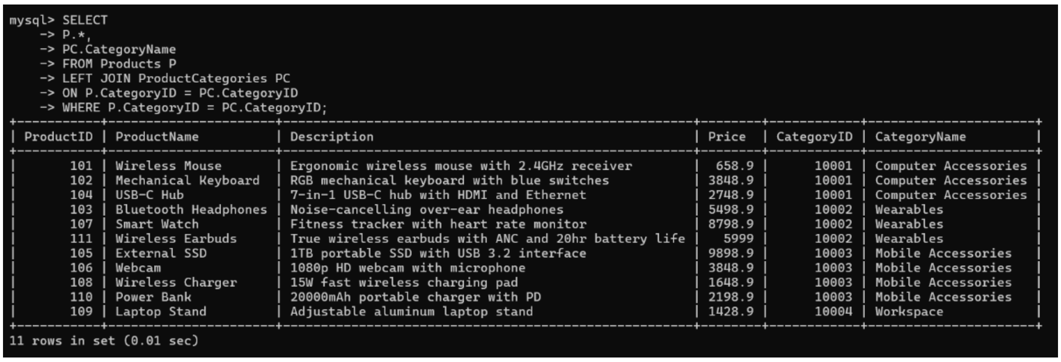
```
mysql> -- Introducing CategoryID from Products as the foreign key to ProductCategories table
mysql> ALTER TABLE Products ADD FOREIGN KEY (CategoryID) REFERENCES ProductCategories(CategoryID);
Query OK, 11 rows affected (0.06 sec)
Records: 11  Duplicates: 0  Warnings: 0
```



```
mysql> -- Updating Products Table with the CategoryID according to the product
mysql> UPDATE Products SET CategoryID =
    -> CASE
    -> WHEN ProductID IN (101, 102, 104) THEN 10001
    -> WHEN ProductID IN (103, 107, 111) THEN 10002
    -> WHEN ProductID IN (105, 106, 108, 110) THEN 10003
    -> WHEN ProductID = 109 THEN 10004
    -> ELSE 10005
    -> END;
Query OK, 11 rows affected (0.01 sec)
Rows matched: 11  Changed: 11  Warnings: 0
```

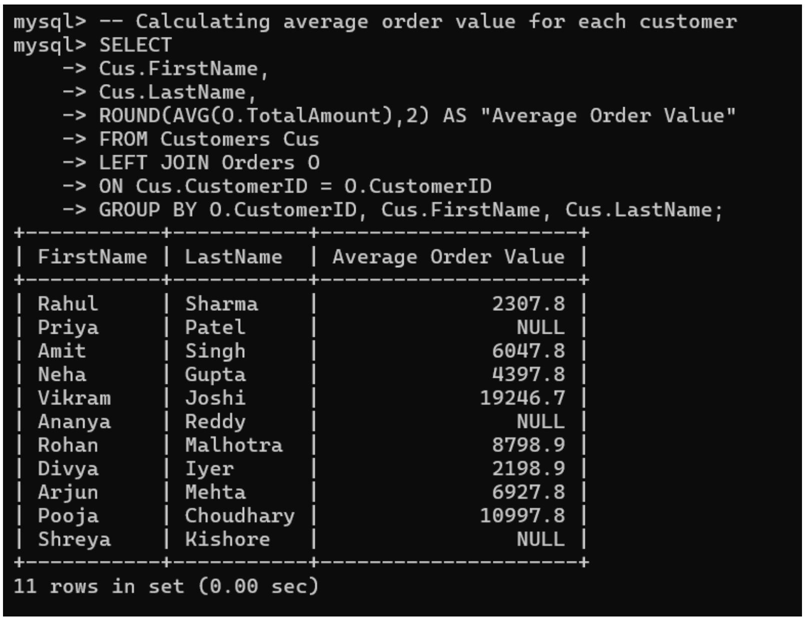


```
mysql> -- Retrieving a list of electronic gadgets along with their corresponding categories
mysql> SELECT * FROM ProductCategories;
+------------+----------------------+----------------------------------+
| CategoryID | CategoryName         | CategoryDesc                     |
+------------+----------------------+----------------------------------+
|      10001 | Computer Accessories | Peripherals for Computers        |
|      10002 | Wearables            | Electronic Devices worn on body  |
|      10003 | Mobile Accessories   | Accessories for Mobile Devices   |
|      10004 | Workspace            | Office and Workspace Equipment   |
|      10005 | Other Electronics    | Miscellaneous electronic gadgets |
+------------+----------------------+----------------------------------+
5 rows in set (0.00 sec)
```



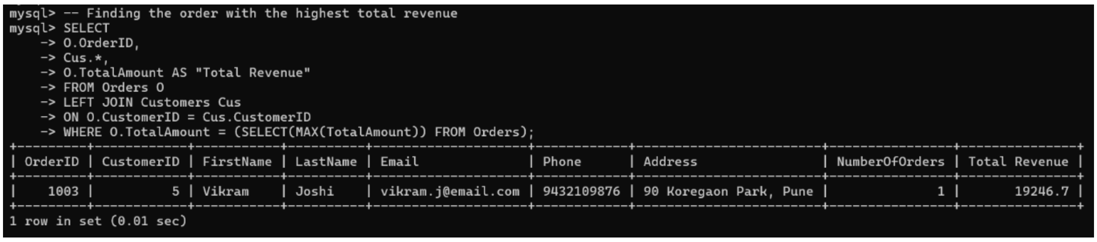
```
mysql> SELECT
    -> P.*,
    -> PC.CategoryName
    -> FROM Products P
    -> LEFT JOIN ProductCategories PC
    -> ON P.CategoryID = PC.CategoryID
    -> WHERE P.CategoryID = PC.CategoryID;
+-----------+----------------------+------------------------------------------------------+--------+------------+----------------------+
| ProductID | ProductName          | Description                                          | Price  | CategoryID | CategoryName         |
+-----------+----------------------+------------------------------------------------------+--------+------------+----------------------+
|       101 | Wireless Mouse       | Ergonomic wireless mouse with 2.4GHz receiver        |  658.9 |      10001 | Computer Accessories |
|       102 | Mechanical Keyboard  | RGB mechanical keyboard with blue switches           | 3848.9 |      10001 | Computer Accessories |
|       104 | USB-C Hub            | 7-in-1 USB-C hub with HDMI and Ethernet              | 2748.9 |      10001 | Computer Accessories |
|       103 | Bluetooth Headphones | Noise-cancelling over-ear headphones                 | 5498.9 |      10002 | Wearables            |
|       107 | Smart Watch          | Fitness tracker with heart rate monitor              | 8798.9 |      10002 | Wearables            |
|       111 | Wireless Earbuds     | True wireless earbuds with ANC and 20hr battery life |   5999 |      10002 | Wearables            |
|       105 | External SSD         | 1TB portable SSD with USB 3.2 interface              | 9898.9 |      10003 | Mobile Accessories   |
|       106 | Webcam               | 1080p HD webcam with microphone                      | 3848.9 |      10003 | Mobile Accessories   |
|       108 | Wireless Charger     | 15W fast wireless charging pad                       | 1648.9 |      10003 | Mobile Accessories   |
|       110 | Power Bank           | 20000mAh portable charger with PD                    | 2198.9 |      10003 | Mobile Accessories   |
|       109 | Laptop Stand         | Adjustable aluminum laptop stand                     | 1428.9 |      10004 | Workspace            |
+-----------+----------------------+------------------------------------------------------+--------+------------+----------------------+
11 rows in set (0.01 sec)
```

6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.



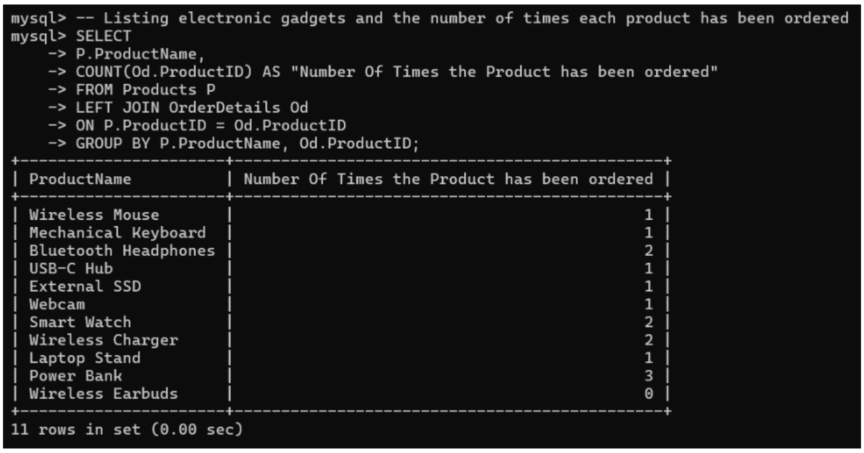
```
mysql> -- Calculating average order value for each customer
mysql> SELECT
    -> Cus.FirstName,
    -> Cus.LastName,
    -> ROUND(AVG(O.TotalAmount),2) AS "Average Order Value"
    -> FROM Customers Cus
    -> LEFT JOIN Orders O
    -> ON Cus.CustomerID = O.CustomerID
    -> GROUP BY O.CustomerID, Cus.FirstName, Cus.LastName;
+-----------+-----------+---------------------+
| FirstName | LastName  | Average Order Value |
+-----------+-----------+---------------------+
| Rahul     | Sharma    |              2307.8 |
| Priya     | Patel     |                NULL |
| Amit      | Singh     |              6047.8 |
| Neha      | Gupta     |              4397.8 |
| Vikram    | Joshi     |             19246.7 |
| Ananya    | Reddy     |                NULL |
| Rohan     | Malhotra  |              8798.9 |
| Divya     | Iyer      |              2198.9 |
| Arjun     | Mehta     |              6927.8 |
| Pooja     | Choudhary |             10997.8 |
| Shreya    | Kishore   |                NULL |
+-----------+-----------+---------------------+
11 rows in set (0.00 sec)
```

7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.



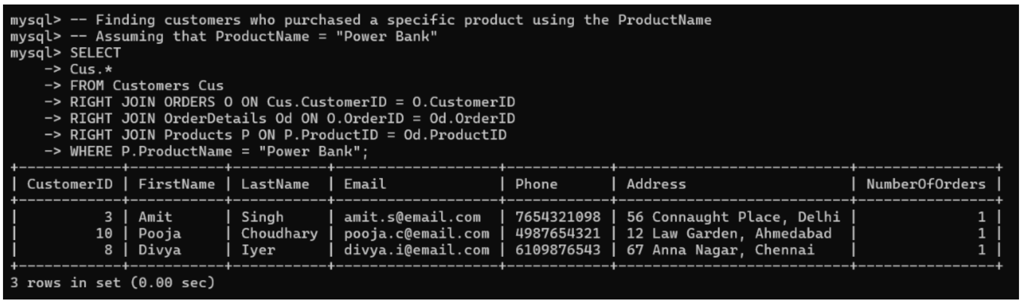
```
mysql> -- Finding the order with the highest total revenue
mysql> SELECT
    -> O.OrderID,
    -> Cus.*,
    -> O.TotalAmount AS "Total Revenue"
    -> FROM Orders O
    -> LEFT JOIN Customers Cus
    -> ON O.CustomerID = Cus.CustomerID
    -> WHERE O.TotalAmount = (SELECT(MAX(TotalAmount)) FROM Orders);
+---------+------------+-----------+----------+--------------------+------------+------------------------+----------------+---------------+
| OrderID | CustomerID | FirstName | LastName | Email              | Phone      | Address                | NumberOfOrders | Total Revenue |
+---------+------------+-----------+----------+--------------------+------------+------------------------+----------------+---------------+
|    1003 |          5 | Vikram    | Joshi    | vikram.j@email.com | 9432109876 | 90 Koregaon Park, Pune |              1 |       19246.7 |
+---------+------------+-----------+----------+--------------------+------------+------------------------+----------------+---------------+
1 row in set (0.01 sec)
```

8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.



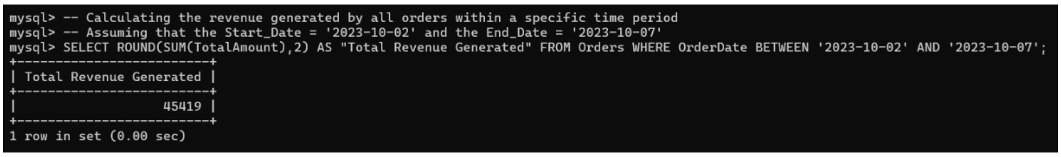
```
mysql> -- Listing electronic gadgets and the number of times each product has been ordered
mysql> SELECT
    -> P.ProductName,
    -> COUNT(Od.ProductID) AS "Number Of Times the Product has been ordered"
    -> FROM Products P
    -> LEFT JOIN OrderDetails Od
    -> ON P.ProductID = Od.ProductID
    -> GROUP BY P.ProductName, Od.ProductID;
+----------------------+----------------------------------------------+
| ProductName          | Number Of Times the Product has been ordered |
+----------------------+----------------------------------------------+
| Wireless Mouse       |                                            1 |
| Mechanical Keyboard  |                                            1 |
| Bluetooth Headphones |                                            2 |
| USB-C Hub            |                                            1 |
| External SSD         |                                            1 |
| Webcam               |                                            1 |
| Smart Watch          |                                            2 |
| Wireless Charger     |                                            2 |
| Laptop Stand         |                                            1 |
| Power Bank           |                                            3 |
| Wireless Earbuds     |                                            0 |
+----------------------+----------------------------------------------+
11 rows in set (0.00 sec)
```

9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.



```
mysql> -- Finding customers who purchased a specific product using the ProductName
mysql> -- Assuming that ProductName = "Power Bank"
mysql> SELECT
    -> Cus.*
    -> FROM Customers Cus
    -> RIGHT JOIN ORDERS O ON Cus.CustomerID = O.CustomerID
    -> RIGHT JOIN OrderDetails Od ON O.OrderID = Od.OrderID
    -> RIGHT JOIN Products P ON P.ProductID = Od.ProductID
    -> WHERE P.ProductName = "Power Bank";
+------------+-----------+-----------+-------------------+------------+---------------------------+----------------+
| CustomerID | FirstName | LastName  | Email             | Phone      | Address                   | NumberOfOrders |
+------------+-----------+-----------+-------------------+------------+---------------------------+----------------+
|          3 | Amit      | Singh     | amit.s@email.com  | 7654321098 | 56 Connaught Place, Delhi |              1 |
|         10 | Pooja     | Choudhary | pooja.c@email.com | 4987654321 | 12 Law Garden, Ahmedabad  |              1 |
|          8 | Divya     | Iyer      | divya.i@email.com | 6109876543 | 67 Anna Nagar, Chennai    |              1 |
+------------+-----------+-----------+-------------------+------------+---------------------------+----------------+
3 rows in set (0.00 sec)
```

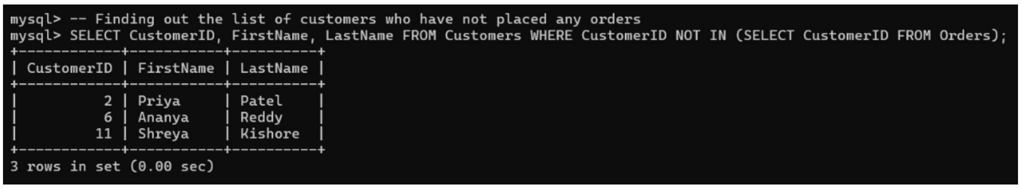
10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.



```
mysql> -- Calculating the revenue generated by all orders within a specific time period
mysql> -- Assuming that the Start_Date = '2023-10-02' and the End_Date = '2023-10-07'
mysql> SELECT ROUND(SUM(TotalAmount),2) AS "Total Revenue Generated" FROM Orders WHERE OrderDate BETWEEN '2023-10-02' AND '2023-10-07';
+-------------------------+
| Total Revenue Generated |
+-------------------------+
|                   45419 |
+-------------------------+
1 row in set (0.00 sec)
```

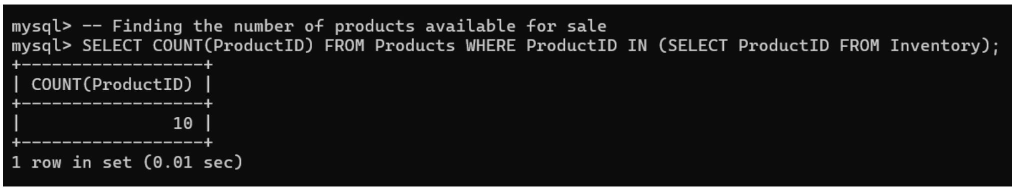
Task 4: Subquery and its types

1. Write an SQL query to find out which customers have not placed any orders.



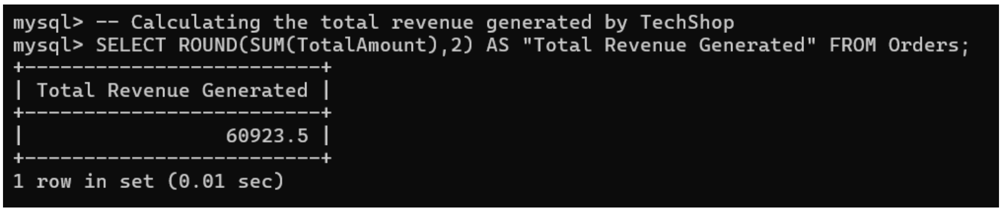
```
mysql> -- Finding out the list of customers who have not placed any orders
mysql> SELECT CustomerID, FirstName, LastName FROM Customers WHERE CustomerID NOT IN (SELECT CustomerID FROM Orders);
+------------+-----------+----------+
| CustomerID | FirstName | LastName |
+------------+-----------+----------+
|          2 | Priya     | Patel    |
|          6 | Ananya    | Reddy    |
|         11 | Shreya    | Kishore  |
+------------+-----------+----------+
3 rows in set (0.00 sec)
```

2. Write an SQL query to find the total number of products available for sale.



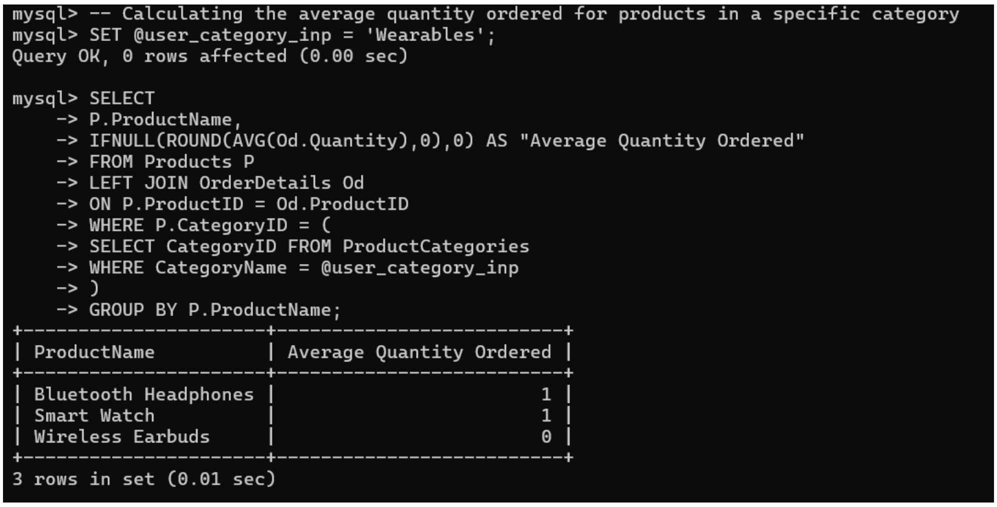
```
mysql> -- Finding the number of products available for sale
mysql> SELECT COUNT(ProductID) FROM Products WHERE ProductID IN (SELECT ProductID FROM Inventory);
+------------------+
| COUNT(ProductID) |
+------------------+
|               10 |
+------------------+
1 row in set (0.01 sec)
```

3. Write an SQL query to calculate the total revenue generated by TechShop.



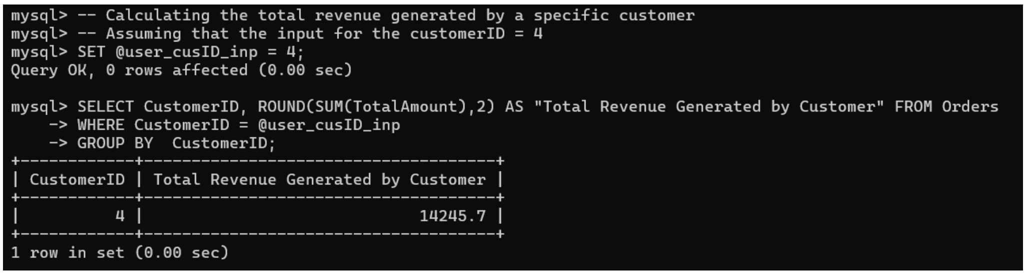
```
mysql> -- Calculating the total revenue generated by TechShop
mysql> SELECT ROUND(SUM(TotalAmount),2) AS "Total Revenue Generated" FROM Orders;
+-------------------------+
| Total Revenue Generated |
+-------------------------+
|                 60923.5 |
+-------------------------+
1 row in set (0.01 sec)
```

4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.



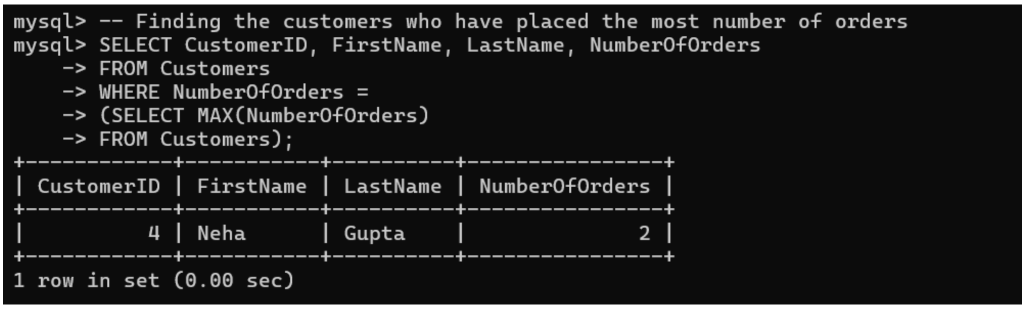
```
mysql> -- Calculating the average quantity ordered for products in a specific category
mysql> SET @user_category_inp = 'Wearables';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT
    -> P.ProductName,
    -> IFNULL(ROUND(AVG(Od.Quantity),0),0) AS "Average Quantity Ordered"
    -> FROM Products P
    -> LEFT JOIN OrderDetails Od
    -> ON P.ProductID = Od.ProductID
    -> WHERE P.CategoryID = (
    -> SELECT CategoryID FROM ProductCategories
    -> WHERE CategoryName = @user_category_inp
    -> )
    -> GROUP BY P.ProductName;
+---------------------+--------------------------+
| ProductName         | Average Quantity Ordered |
+---------------------+--------------------------+
| Bluetooth Headphones |                        1 |
| Smart Watch         |                        1 |
| Wireless Earbuds    |                        0 |
+---------------------+--------------------------+
3 rows in set (0.01 sec)
```

5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.



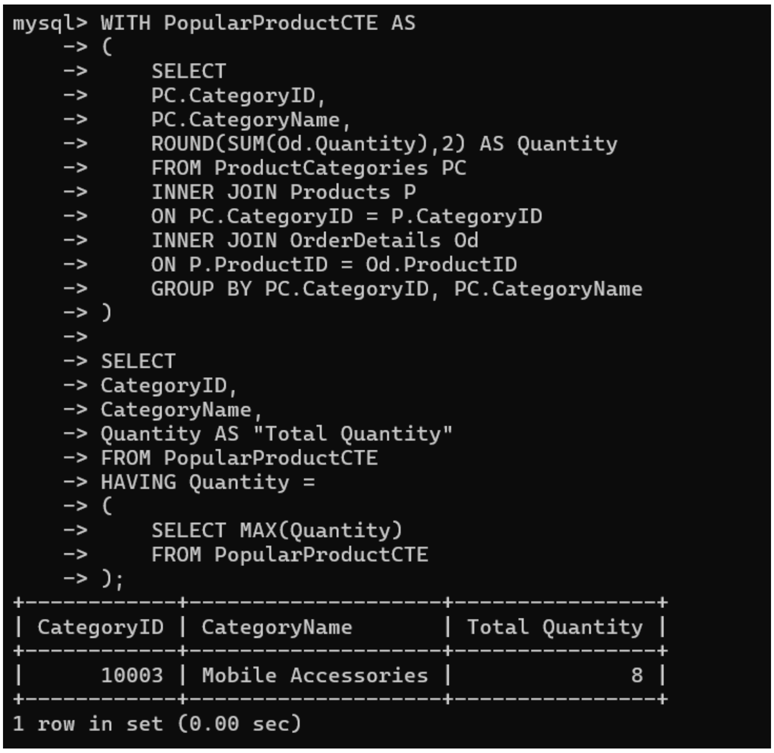
```
mysql> -- Calculating the total revenue generated by a specific customer
mysql> -- Assuming that the input for the customerID = 4
mysql> SET @user_cusID_inp = 4;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT CustomerID, ROUND(SUM(TotalAmount),2) AS "Total Revenue Generated by Customer" FROM Orders
    -> WHERE CustomerID = @user_cusID_inp
    -> GROUP BY  CustomerID;
+------------+-------------------------------------+
| CustomerID | Total Revenue Generated by Customer |
+------------+-------------------------------------+
|          4 |                             14245.7 |
+------------+-------------------------------------+
1 row in set (0.00 sec)
```

6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.



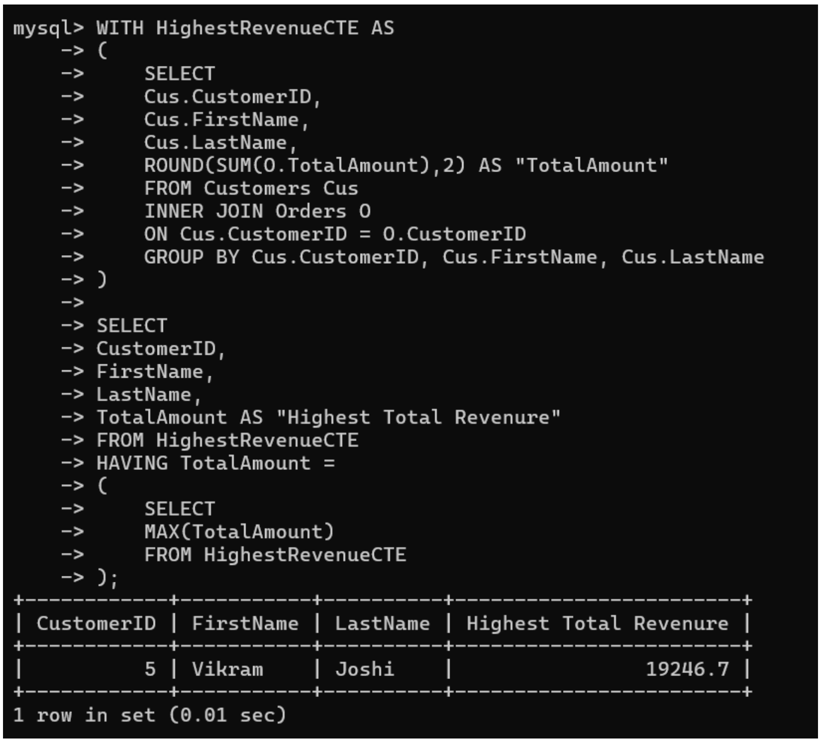
```
mysql> -- Finding the customers who have placed the most number of orders
mysql> SELECT CustomerID, FirstName, LastName, NumberOfOrders
    -> FROM Customers
    -> WHERE NumberOfOrders =
    -> (SELECT MAX(NumberOfOrders)
    -> FROM Customers);
+------------+-----------+----------+----------------+
| CustomerID | FirstName | LastName | NumberOfOrders |
+------------+-----------+----------+----------------+
|          4 | Neha      | Gupta    |              2 |
+------------+-----------+----------+----------------+
1 row in set (0.00 sec)
```

7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.



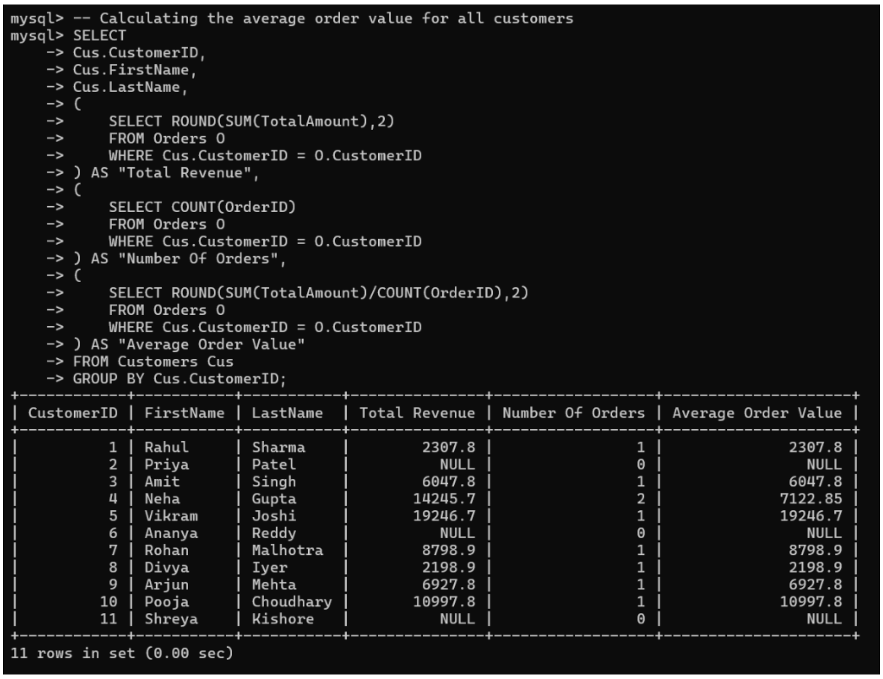
```
mysql> WITH PopularProductCTE AS
    -> (
    ->     SELECT
    ->     PC.CategoryID,
    ->     PC.CategoryName,
    ->     ROUND(SUM(Od.Quantity),2) AS Quantity
    ->     FROM ProductCategories PC
    ->     INNER JOIN Products P
    ->     ON PC.CategoryID = P.CategoryID
    ->     INNER JOIN OrderDetails Od
    ->     ON P.ProductID = Od.ProductID
    ->     GROUP BY PC.CategoryID, PC.CategoryName
    -> )
    ->
    -> SELECT
    -> CategoryID,
    -> CategoryName,
    -> Quantity AS "Total Quantity"
    -> FROM PopularProductCTE
    -> HAVING Quantity =
    -> (
    ->     SELECT MAX(Quantity)
    ->     FROM PopularProductCTE
    -> );
+------------+--------------------+----------------+
| CategoryID | CategoryName       | Total Quantity |
+------------+--------------------+----------------+
|      10003 | Mobile Accessories |              8 |
+------------+--------------------+----------------+
1 row in set (0.00 sec)
```

8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.



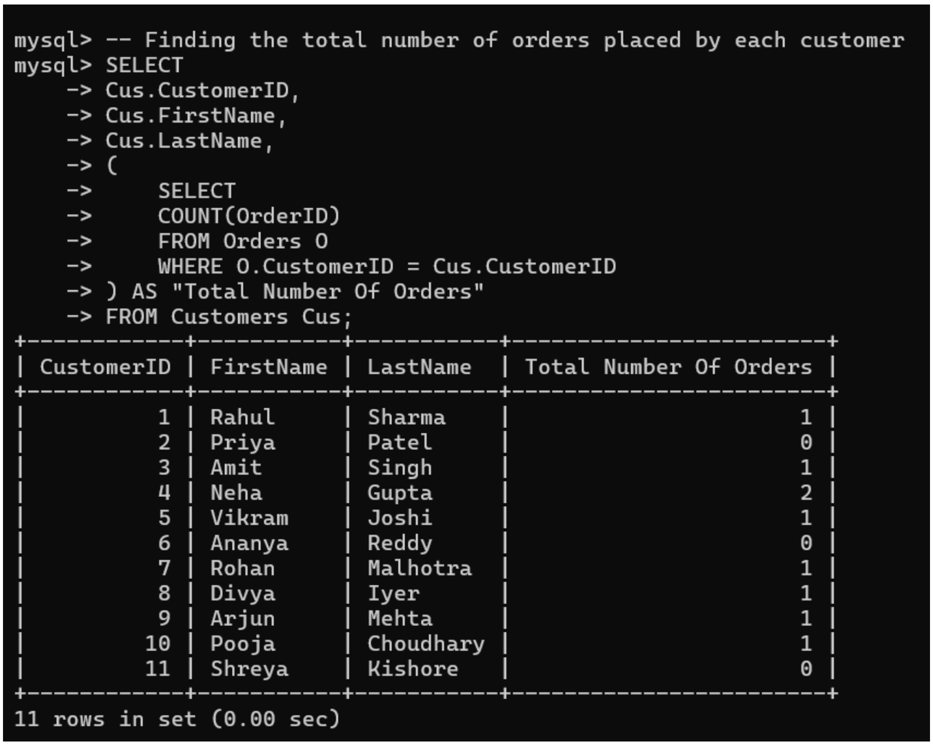
```
mysql> WITH HighestRevenueCTE AS
    -> (
    ->     SELECT
    ->     Cus.CustomerID,
    ->     Cus.FirstName,
    ->     Cus.LastName,
    ->     ROUND(SUM(O.TotalAmount),2) AS "TotalAmount"
    ->     FROM Customers Cus
    ->     INNER JOIN Orders O
    ->     ON Cus.CustomerID = O.CustomerID
    ->     GROUP BY Cus.CustomerID, Cus.FirstName, Cus.LastName
    -> )
    ->
    -> SELECT
    -> CustomerID,
    -> FirstName,
    -> LastName,
    -> TotalAmount AS "Highest Total Revenure"
    -> FROM HighestRevenueCTE
    -> HAVING TotalAmount =
    -> (
    ->     SELECT
    ->     MAX(TotalAmount)
    ->     FROM HighestRevenueCTE
    -> );
+------------+-----------+----------+------------------------+
| CustomerID | FirstName | LastName | Highest Total Revenure |
+------------+-----------+----------+------------------------+
|          5 | Vikram    | Joshi    |                19246.7 |
+------------+-----------+----------+------------------------+
1 row in set (0.01 sec)
```

9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.



```
mysql> -- Calculating the average order value for all customers
mysql> SELECT
    -> Cus.CustomerID,
    -> Cus.FirstName,
    -> Cus.LastName,
    -> (
    ->     SELECT ROUND(SUM(TotalAmount),2)
    ->     FROM Orders O
    ->     WHERE Cus.CustomerID = O.CustomerID
    -> ) AS "Total Revenue",
    -> (
    ->     SELECT COUNT(OrderID)
    ->     FROM Orders O
    ->     WHERE Cus.CustomerID = O.CustomerID
    -> ) AS "Number Of Orders",
    -> (
    ->     SELECT ROUND(SUM(TotalAmount)/COUNT(OrderID),2)
    ->     FROM Orders O
    ->     WHERE Cus.CustomerID = O.CustomerID
    -> ) AS "Average Order Value"
    -> FROM Customers Cus
    -> GROUP BY Cus.CustomerID;
+------------+-----------+-----------+---------------+------------------+---------------------+
| CustomerID | FirstName | LastName  | Total Revenue | Number Of Orders | Average Order Value |
+------------+-----------+-----------+---------------+------------------+---------------------+
|          1 | Rahul     | Sharma    |        2307.8 |                1 |              2307.8 |
|          2 | Priya     | Patel     |          NULL |                0 |                NULL |
|          3 | Amit      | Singh     |        6047.8 |                1 |              6047.8 |
|          4 | Neha      | Gupta     |       14245.7 |                2 |             7122.85 |
|          5 | Vikram    | Joshi     |       19246.7 |                1 |             19246.7 |
|          6 | Ananya    | Reddy     |          NULL |                0 |                NULL |
|          7 | Rohan     | Malhotra  |        8798.9 |                1 |              8798.9 |
|          8 | Divya     | Iyer      |        2198.9 |                1 |              2198.9 |
|          9 | Arjun     | Mehta     |        6927.8 |                1 |              6927.8 |
|         10 | Pooja     | Choudhary |       10997.8 |                1 |             10997.8 |
|         11 | Shreya    | Kishore   |          NULL |                0 |                NULL |
+------------+-----------+-----------+---------------+------------------+---------------------+
11 rows in set (0.00 sec)
```

10. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.



```
mysql> -- Finding the total number of orders placed by each customer
mysql> SELECT
    -> Cus.CustomerID,
    -> Cus.FirstName,
    -> Cus.LastName,
    -> (
    ->     SELECT
    ->     COUNT(OrderID)
    ->     FROM Orders O
    ->     WHERE O.CustomerID = Cus.CustomerID
    -> ) AS "Total Number Of Orders"
    -> FROM Customers Cus;
+------------+-----------+-----------+------------------------+
| CustomerID | FirstName | LastName  | Total Number Of Orders |
+------------+-----------+-----------+------------------------+
|          1 | Rahul     | Sharma    |                      1 |
|          2 | Priya     | Patel     |                      0 |
|          3 | Amit      | Singh     |                      1 |
|          4 | Neha      | Gupta     |                      2 |
|          5 | Vikram    | Joshi     |                      1 |
|          6 | Ananya    | Reddy     |                      0 |
|          7 | Rohan     | Malhotra  |                      1 |
|          8 | Divya     | Iyer      |                      1 |
|          9 | Arjun     | Mehta     |                      1 |
|         10 | Pooja     | Choudhary |                      1 |
|         11 | Shreya    | Kishore   |                      0 |
+------------+-----------+-----------+------------------------+
11 rows in set (0.00 sec)
```