Assignment – 1

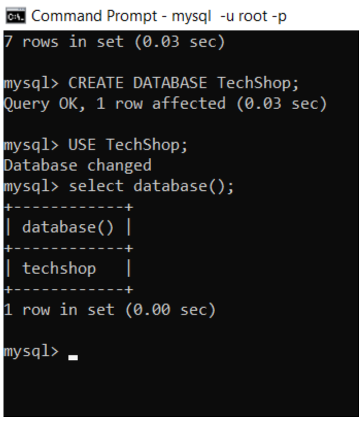
Task:1. Database Design:

1. Create the database named "TechShop"

Ans.

CREATE DATABASE TechShop;

USE TechShop;



Command Prompt - mysql -u root -p

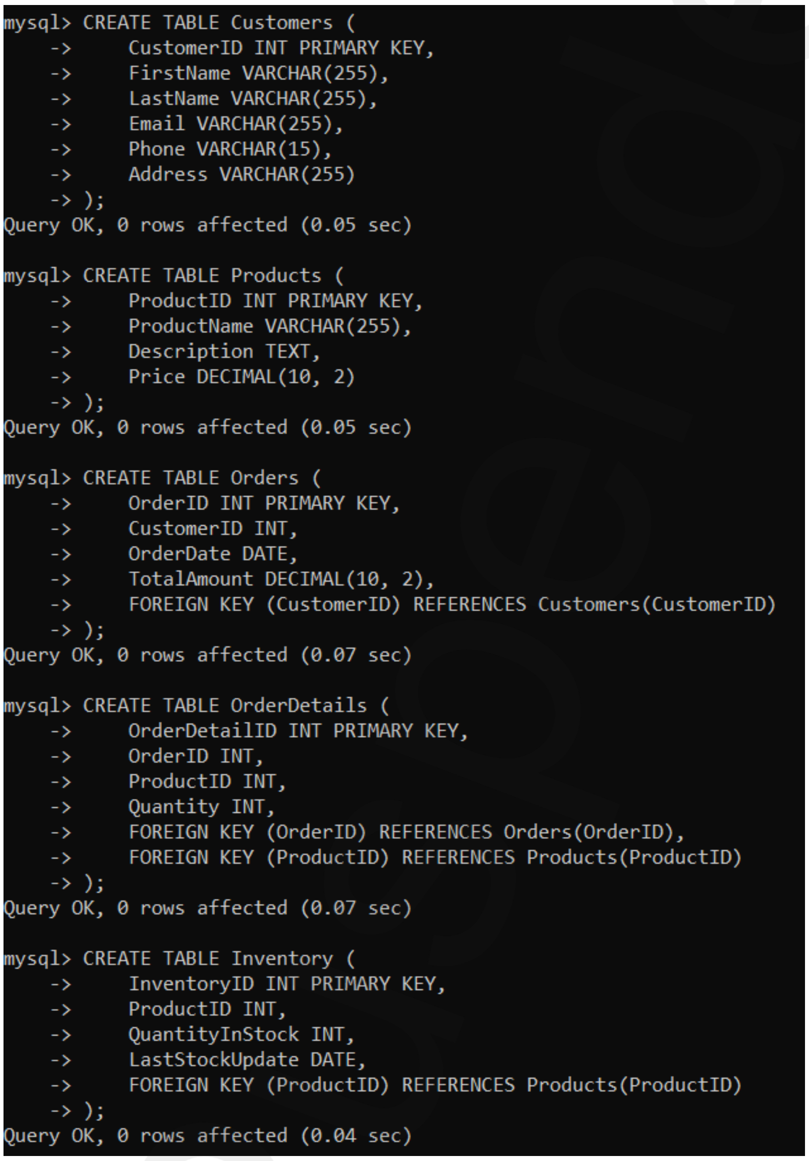
```
7 rows in set (0.03 sec)

mysql> CREATE DATABASE TechShop;
Query OK, 1 row affected (0.03 sec)

mysql> USE TechShop;
Database changed
mysql> select database();
+------------+
| database() |
+------------+
| techshop   |
+------------+
1 row in set (0.00 sec)

mysql>
```

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

Ans.

```
-- Customers Table

CREATE TABLE Customers (

    CustomerID INT PRIMARY KEY,

    FirstName VARCHAR(255),

    LastName VARCHAR(255),

    Email VARCHAR(255),

    Phone VARCHAR(15),

    Address VARCHAR(255)

);
```

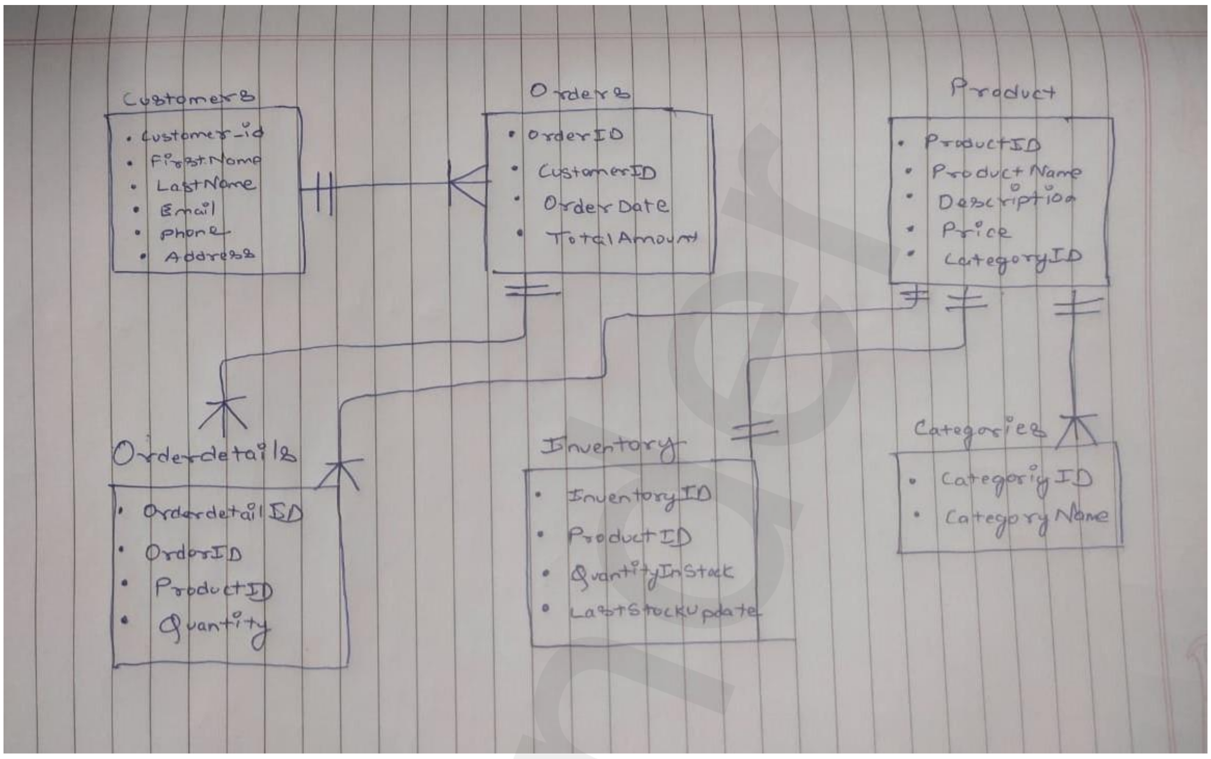
```
-- Products Table
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(255),
    Description TEXT,
    Price DECIMAL(10, 2)
);

-- Orders Table
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

-- OrderDetails Table
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

-- Inventory Table
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,

    ProductID INT,

    QuantityInStock INT,

    LastStockUpdate DATE,

    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);
```



```
mysql> CREATE TABLE Customers (
    ->     CustomerID INT PRIMARY KEY,
    ->     FirstName VARCHAR(255),
    ->     LastName VARCHAR(255),
    ->     Email VARCHAR(255),
    ->     Phone VARCHAR(15),
    ->     Address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Products (
    ->     ProductID INT PRIMARY KEY,
    ->     ProductName VARCHAR(255),
    ->     Description TEXT,
    ->     Price DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Orders (
    ->     OrderID INT PRIMARY KEY,
    ->     CustomerID INT,
    ->     OrderDate DATE,
    ->     TotalAmount DECIMAL(10, 2),
    ->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE OrderDetails (
    ->     OrderDetailID INT PRIMARY KEY,
    ->     OrderID INT,
    ->     ProductID INT,
    ->     Quantity INT,
    ->     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE Inventory (
    ->     InventoryID INT PRIMARY KEY,
    ->     ProductID INT,
    ->     QuantityInStock INT,
    ->     LastStockUpdate DATE,
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

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

Ans.



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

Ans. Primary Key constraints are already added in the table definitions.

Foreign Key constraints are also added in the table definition

5. 5. Insert at least 10 sample records into each of the following tables:
   a. Customers
   b. Product
   c. Orders
   d. OrderDetatials
   e. Inventory

Ans.

-- Inserting sample records into the Customers table

INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)

VALUES

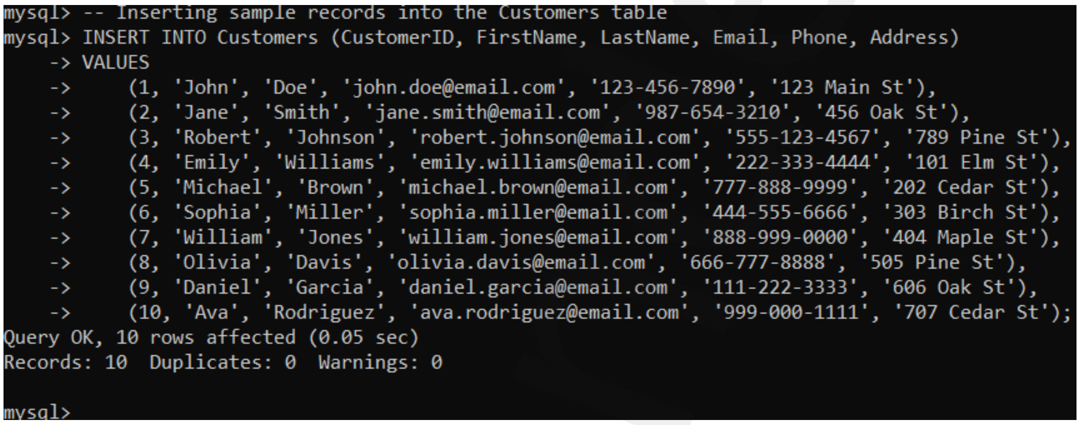
(1, 'John', 'Doe', 'john.doe@email.com', '123-456-7890', '123 Main St'),

(2, 'Jane', 'Smith', 'jane.smith@email.com', '987-654-3210', '456 Oak St'),

(3, 'Robert', 'Johnson', 'robert.johnson@email.com', '555-123-4567', '789 Pine St'),

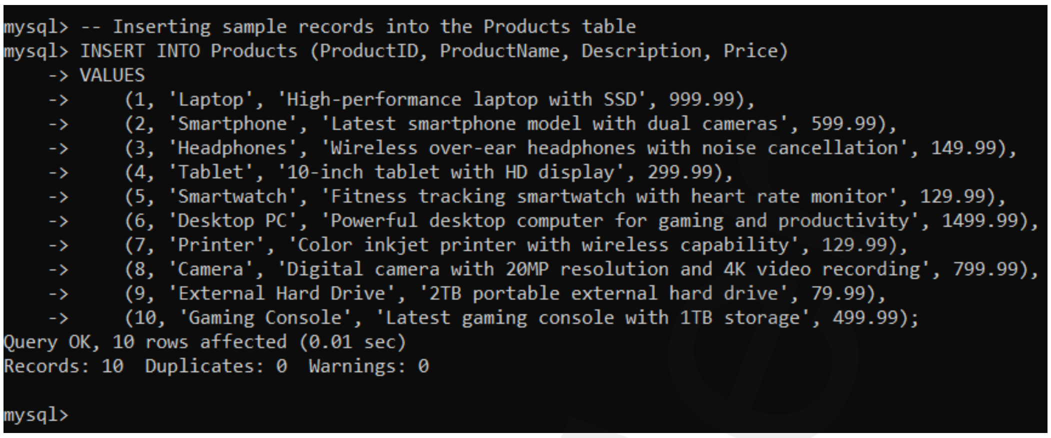
(4, 'Emily', 'Williams', 'emily.williams@email.com', '222-333-4444', '101 Elm St'),

```
    (5, 'Michael', 'Brown', 'michael.brown@email.com', '777-888-9999', '202 Cedar St'),
    (6, 'Sophia', 'Miller', 'sophia.miller@email.com', '444-555-6666', '303 Birch St'),
    (7, 'William', 'Jones', 'william.jones@email.com', '888-999-0000', '404 Maple St'),
    (8, 'Olivia', 'Davis', 'olivia.davis@email.com', '666-777-8888', '505 Pine St'),
    (9, 'Daniel', 'Garcia', 'daniel.garcia@email.com', '111-222-3333', '606 Oak St'),
    (10, 'Ava', 'Rodriguez', 'ava.rodriguez@email.com', '999-000-1111', '707 Cedar St');
```



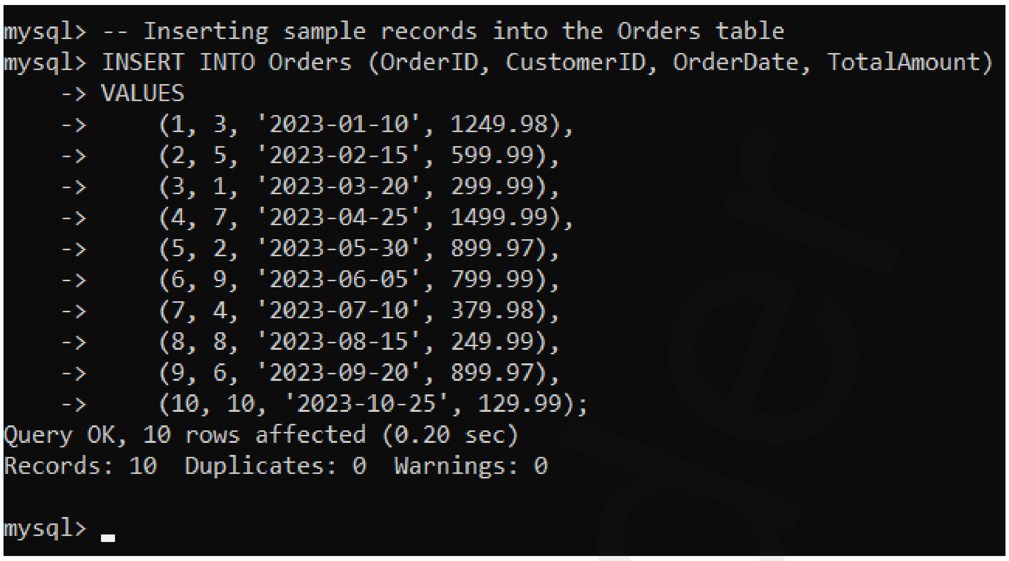
```
mysql> -- Inserting sample records into the Customers table
mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
    -> VALUES
    ->     (1, 'John', 'Doe', 'john.doe@email.com', '123-456-7890', '123 Main St'),
    ->     (2, 'Jane', 'Smith', 'jane.smith@email.com', '987-654-3210', '456 Oak St'),
    ->     (3, 'Robert', 'Johnson', 'robert.johnson@email.com', '555-123-4567', '789 Pine St'),
    ->     (4, 'Emily', 'Williams', 'emily.williams@email.com', '222-333-4444', '101 Elm St'),
    ->     (5, 'Michael', 'Brown', 'michael.brown@email.com', '777-888-9999', '202 Cedar St'),
    ->     (6, 'Sophia', 'Miller', 'sophia.miller@email.com', '444-555-6666', '303 Birch St'),
    ->     (7, 'William', 'Jones', 'william.jones@email.com', '888-999-0000', '404 Maple St'),
    ->     (8, 'Olivia', 'Davis', 'olivia.davis@email.com', '666-777-8888', '505 Pine St'),
    ->     (9, 'Daniel', 'Garcia', 'daniel.garcia@email.com', '111-222-3333', '606 Oak St'),
    ->     (10, 'Ava', 'Rodriguez', 'ava.rodriguez@email.com', '999-000-1111', '707 Cedar St');
Query OK, 10 rows affected (0.05 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
```

```
-- Inserting sample records into the Products table
INSERT INTO Products (ProductID, ProductName, Description, Price)
VALUES
    (1, 'Laptop', 'High-performance laptop with SSD', 999.99),
    (2, 'Smartphone', 'Latest smartphone model with dual cameras', 599.99),
    (3, 'Headphones', 'Wireless over-ear headphones with noise cancellation', 149.99),
    (4, 'Tablet', '10-inch tablet with HD display', 299.99),
    (5, 'Smartwatch', 'Fitness tracking smartwatch with heart rate monitor', 129.99),
    (6, 'Desktop PC', 'Powerful desktop computer for gaming and productivity', 1499.99),
    (7, 'Printer', 'Color inkjet printer with wireless capability', 129.99),
    (8, 'Camera', 'Digital camera with 20MP resolution and 4K video recording', 799.99),
    (9, 'External Hard Drive', '2TB portable external hard drive', 79.99),
    (10, 'Gaming Console', 'Latest gaming console with 1TB storage', 499.99);
```



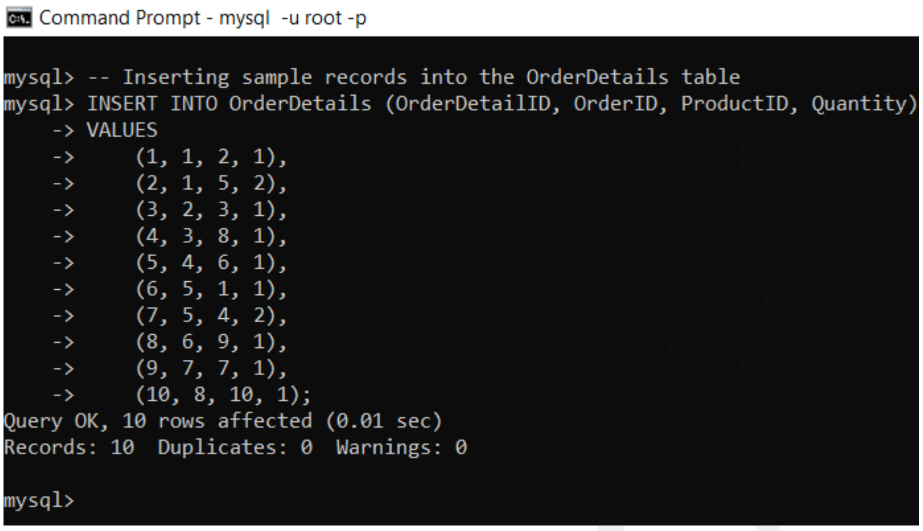
```
mysql> -- Inserting sample records into the Products table
mysql> INSERT INTO Products (ProductID, ProductName, Description, Price)
    -> VALUES
    ->     (1, 'Laptop', 'High-performance laptop with SSD', 999.99),
    ->     (2, 'Smartphone', 'Latest smartphone model with dual cameras', 599.99),
    ->     (3, 'Headphones', 'Wireless over-ear headphones with noise cancellation', 149.99),
    ->     (4, 'Tablet', '10-inch tablet with HD display', 299.99),
    ->     (5, 'Smartwatch', 'Fitness tracking smartwatch with heart rate monitor', 129.99),
    ->     (6, 'Desktop PC', 'Powerful desktop computer for gaming and productivity', 1499.99),
    ->     (7, 'Printer', 'Color inkjet printer with wireless capability', 129.99),
    ->     (8, 'Camera', 'Digital camera with 20MP resolution and 4K video recording', 799.99),
    ->     (9, 'External Hard Drive', '2TB portable external hard drive', 79.99),
    ->     (10, 'Gaming Console', 'Latest gaming console with 1TB storage', 499.99);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
```

```
-- Inserting sample records into the Orders table

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)

VALUES

   (1, 3, '2023-01-10', 1249.98),

   (2, 5, '2023-02-15', 599.99),

   (3, 1, '2023-03-20', 299.99),

   (4, 7, '2023-04-25', 1499.99),

   (5, 2, '2023-05-30', 899.97),

   (6, 9, '2023-06-05', 799.99),

   (7, 4, '2023-07-10', 379.98),

   (8, 8, '2023-08-15', 249.99),

   (9, 6, '2023-09-20', 899.97),

   (10, 10, '2023-10-25', 129.99);
```



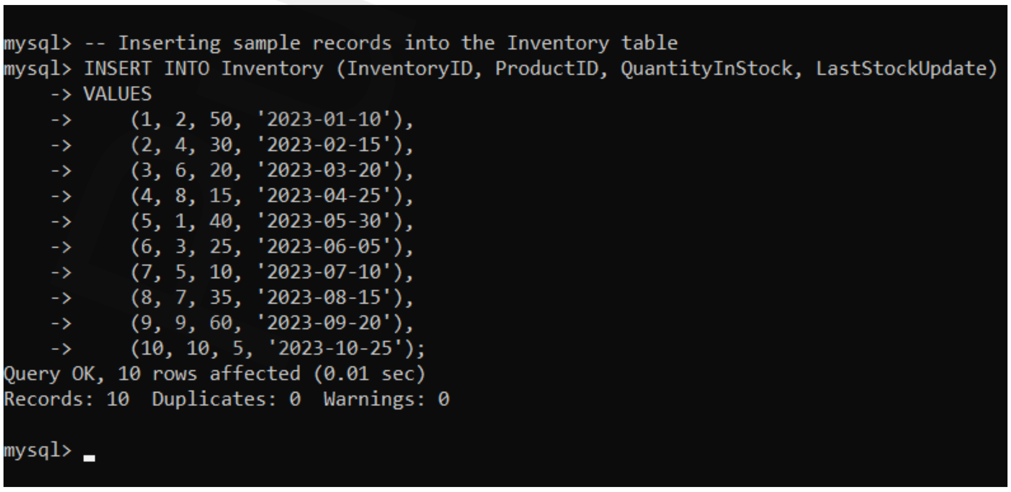
```
mysql> -- Inserting sample records into the Orders table
mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
    -> VALUES
    ->     (1, 3, '2023-01-10', 1249.98),
    ->     (2, 5, '2023-02-15', 599.99),
    ->     (3, 1, '2023-03-20', 299.99),
    ->     (4, 7, '2023-04-25', 1499.99),
    ->     (5, 2, '2023-05-30', 899.97),
    ->     (6, 9, '2023-06-05', 799.99),
    ->     (7, 4, '2023-07-10', 379.98),
    ->     (8, 8, '2023-08-15', 249.99),
    ->     (9, 6, '2023-09-20', 899.97),
    ->     (10, 10, '2023-10-25', 129.99);
Query OK, 10 rows affected (0.20 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> _
```

```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)

VALUES

    (1, 1, 2, 1),

    (2, 1, 5, 2),

    (3, 2, 3, 1),

    (4, 3, 8, 1),

    (5, 4, 6, 1),

    (6, 5, 1, 1),

    (7, 5, 4, 2),

    (8, 6, 9, 1),

    (9, 7, 7, 1),

    (10, 8, 10, 1);
```


Command Prompt - mysql -u root -p

```
mysql> -- Inserting sample records into the OrderDetails table
mysql> INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
    -> VALUES
    ->     (1, 1, 2, 1),
    ->     (2, 1, 5, 2),
    ->     (3, 2, 3, 1),
    ->     (4, 3, 8, 1),
    ->     (5, 4, 6, 1),
    ->     (6, 5, 1, 1),
    ->     (7, 5, 4, 2),
    ->     (8, 6, 9, 1),
    ->     (9, 7, 7, 1),
    ->     (10, 8, 10, 1);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
```

```
-- Inserting sample records into the Inventory table

INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)

VALUES

    (1, 2, 50, '2023-01-10'),

    (2, 4, 30, '2023-02-15'),

    (3, 6, 20, '2023-03-20'),

    (4, 8, 15, '2023-04-25'),

    (5, 1, 40, '2023-05-30'),

    (6, 3, 25, '2023-06-05'),

    (7, 5, 10, '2023-07-10'),

    (8, 7, 35, '2023-08-15'),

    (9, 9, 60, '2023-09-20'),

    (10, 10, 5, '2023-10-25');
```



```
mysql> -- Inserting sample records into the Inventory table
mysql> INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)
    -> VALUES
    ->     (1, 2, 50, '2023-01-10'),
    ->     (2, 4, 30, '2023-02-15'),
    ->     (3, 6, 20, '2023-03-20'),
    ->     (4, 8, 15, '2023-04-25'),
    ->     (5, 1, 40, '2023-05-30'),
    ->     (6, 3, 25, '2023-06-05'),
    ->     (7, 5, 10, '2023-07-10'),
    ->     (8, 7, 35, '2023-08-15'),
    ->     (9, 9, 60, '2023-09-20'),
    ->     (10, 10, 5, '2023-10-25');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
```

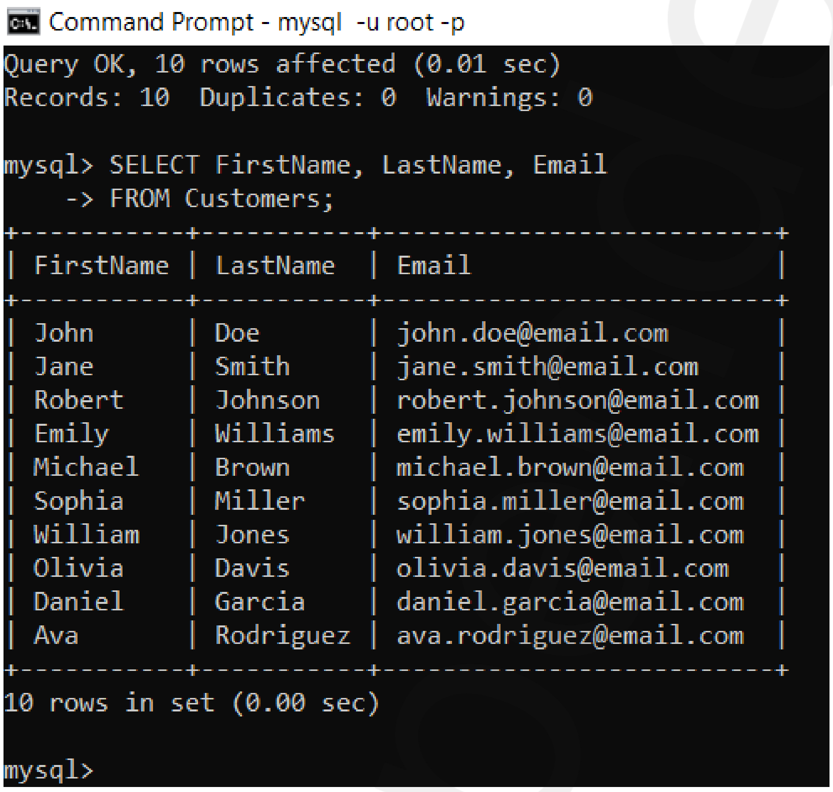
Tasks 2: Select, Where, Between, AND, LIKE:

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

Ans.

SELECT FirstName, LastName, Email

FROM Customers;



Command Prompt - mysql -u root -p

```
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> SELECT FirstName, LastName, Email
    -> FROM Customers;
+-----------+-----------+--------------------------+
| FirstName | LastName  | Email                    |
+-----------+-----------+--------------------------+
| John      | Doe       | john.doe@email.com       |
| Jane      | Smith     | jane.smith@email.com     |
| Robert    | Johnson   | robert.johnson@email.com |
| Emily     | Williams  | emily.williams@email.com |
| Michael   | Brown     | michael.brown@email.com  |
| Sophia    | Miller    | sophia.miller@email.com  |
| William   | Jones     | william.jones@email.com  |
| Olivia    | Davis     | olivia.davis@email.com   |
| Daniel    | Garcia    | daniel.garcia@email.com  |
| Ava       | Rodriguez | ava.rodriguez@email.com  |
+-----------+-----------+--------------------------+
10 rows in set (0.00 sec)

mysql>
```

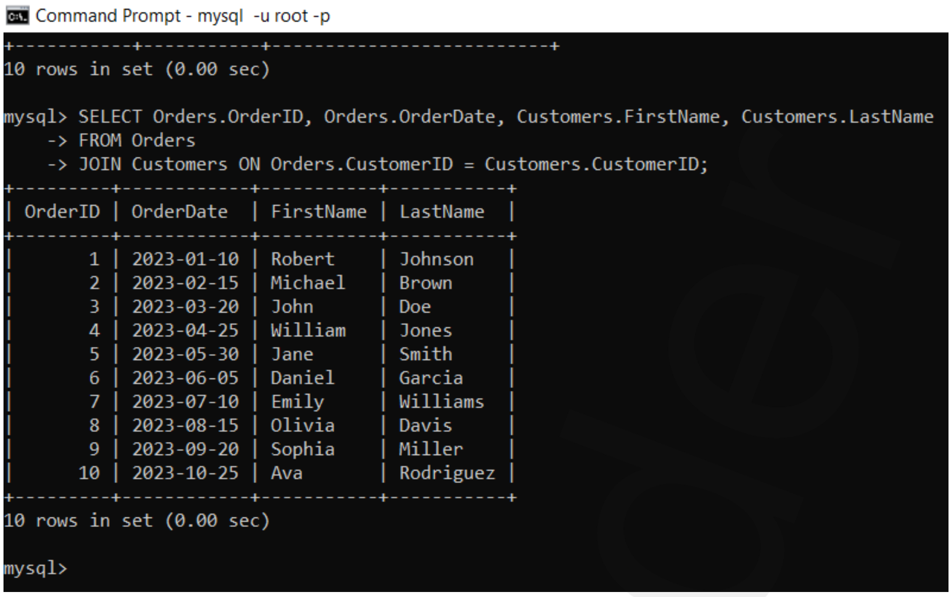
2. Write an SQL query to list all orders with their order dates and corresponding customer names

Ans.

SELECT Orders.OrderID, Orders.OrderDate, Customers.FirstName, Customers.LastName

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID;



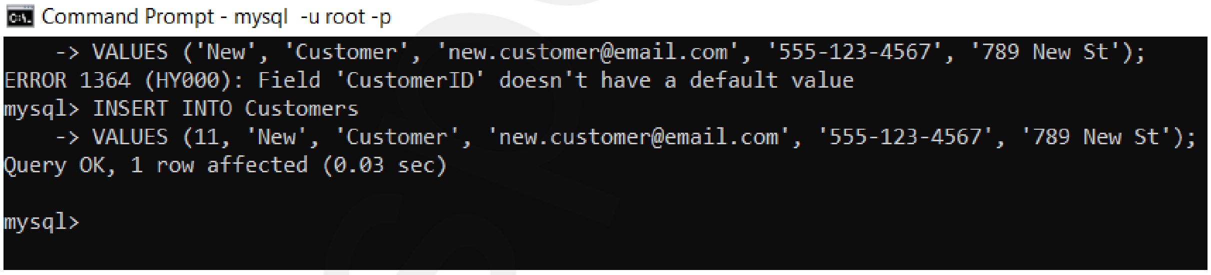
```
Command Prompt - mysql  -u root -p
+-----------+-----------+-------------------------+
10 rows in set (0.00 sec)

mysql> SELECT Orders.OrderID, Orders.OrderDate, Customers.FirstName, Customers.LastName
    -> FROM Orders
    -> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
+---------+------------+-----------+-----------+
| OrderID | OrderDate  | FirstName | LastName  |
+---------+------------+-----------+-----------+
|       1 | 2023-01-10 | Robert    | Johnson   |
|       2 | 2023-02-15 | Michael   | Brown     |
|       3 | 2023-03-20 | John      | Doe       |
|       4 | 2023-04-25 | William   | Jones     |
|       5 | 2023-05-30 | Jane      | Smith     |
|       6 | 2023-06-05 | Daniel    | Garcia    |
|       7 | 2023-07-10 | Emily     | Williams  |
|       8 | 2023-08-15 | Olivia    | Davis     |
|       9 | 2023-09-20 | Sophia    | Miller    |
|      10 | 2023-10-25 | Ava       | Rodriguez |
+---------+------------+-----------+-----------+
10 rows in set (0.00 sec)

mysql>
```

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

Ans.

INSERT INTO Customers (FirstName, LastName, Email, Phone, Address)

VALUES ('New', 'Customer', 'new.customer@email.com', '555-123-4567', '789 New St');



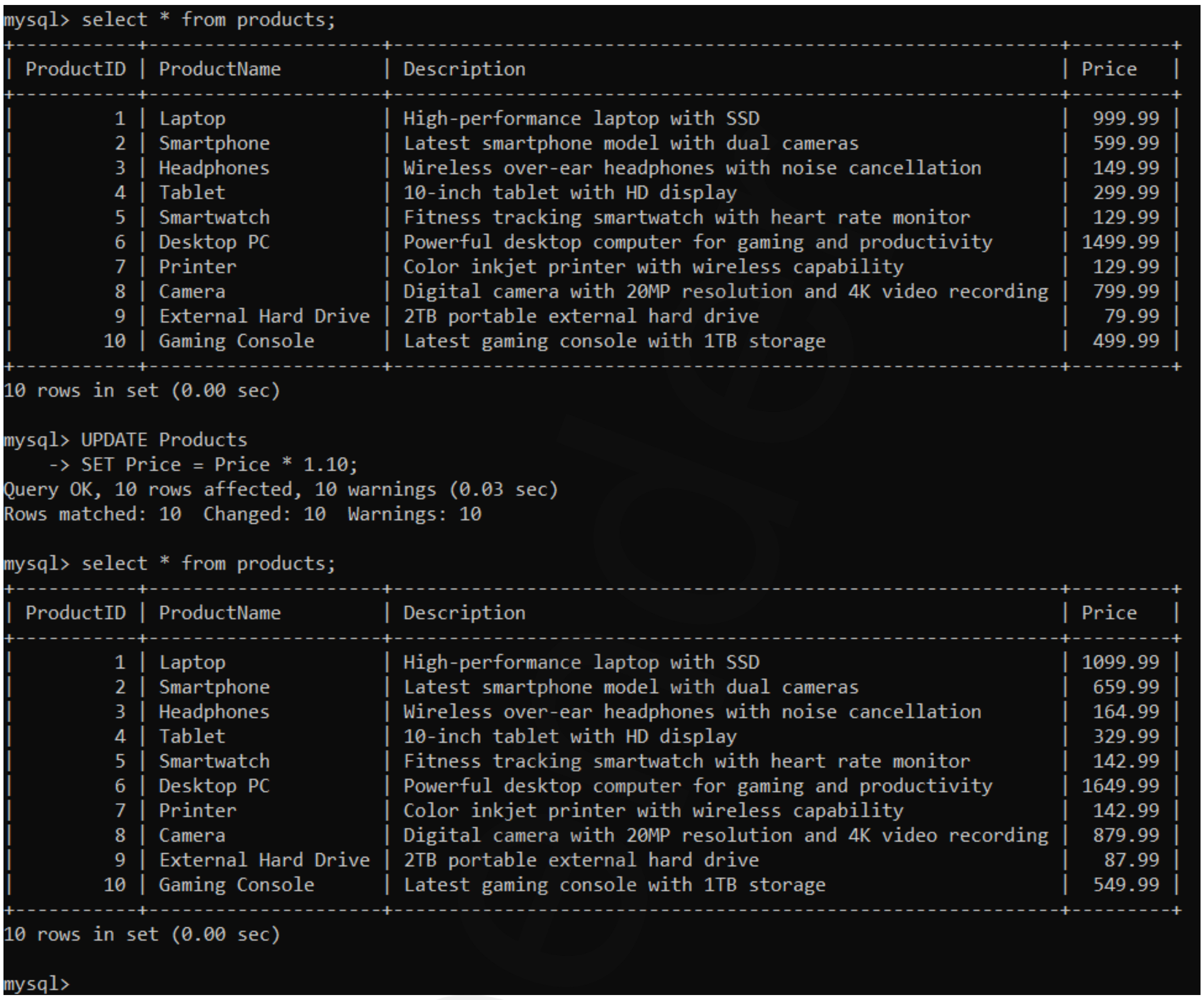
```
Command Prompt - mysql  -u root -p
    -> VALUES ('New', 'Customer', 'new.customer@email.com', '555-123-4567', '789 New St');
ERROR 1364 (HY000): Field 'CustomerID' doesn't have a default value
mysql> INSERT INTO Customers
    -> VALUES (11, 'New', 'Customer', 'new.customer@email.com', '555-123-4567', '789 New St');
Query OK, 1 row affected (0.03 sec)

mysql>
```

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

Ans.

UPDATE Products

SET Price = Price * 1.10;


```
mysql> select * from products;
+-----------+---------------------+------------------------------------------------------------+---------+
| ProductID | ProductName         | Description                                                | Price   |
+-----------+---------------------+------------------------------------------------------------+---------+
|         1 | Laptop              | High-performance laptop with SSD                           |  999.99 |
|         2 | Smartphone          | Latest smartphone model with dual cameras                  |  599.99 |
|         3 | Headphones          | Wireless over-ear headphones with noise cancellation       |  149.99 |
|         4 | Tablet              | 10-inch tablet with HD display                             |  299.99 |
|         5 | Smartwatch          | Fitness tracking smartwatch with heart rate monitor        |  129.99 |
|         6 | Desktop PC          | Powerful desktop computer for gaming and productivity      | 1499.99 |
|         7 | Printer             | Color inkjet printer with wireless capability              |  129.99 |
|         8 | Camera              | Digital camera with 20MP resolution and 4K video recording |  799.99 |
|         9 | External Hard Drive | 2TB portable external hard drive                           |   79.99 |
|        10 | Gaming Console      | Latest gaming console with 1TB storage                     |  499.99 |
+-----------+---------------------+------------------------------------------------------------+---------+
10 rows in set (0.00 sec)

mysql> UPDATE Products
    -> SET Price = Price * 1.10;
Query OK, 10 rows affected, 10 warnings (0.03 sec)
Rows matched: 10  Changed: 10  Warnings: 10

mysql> select * from products;
+-----------+---------------------+------------------------------------------------------------+---------+
| ProductID | ProductName         | Description                                                | Price   |
+-----------+---------------------+------------------------------------------------------------+---------+
|         1 | Laptop              | High-performance laptop with SSD                           | 1099.99 |
|         2 | Smartphone          | Latest smartphone model with dual cameras                  |  659.99 |
|         3 | Headphones          | Wireless over-ear headphones with noise cancellation       |  164.99 |
|         4 | Tablet              | 10-inch tablet with HD display                             |  329.99 |
|         5 | Smartwatch          | Fitness tracking smartwatch with heart rate monitor        |  142.99 |
|         6 | Desktop PC          | Powerful desktop computer for gaming and productivity      | 1649.99 |
|         7 | Printer             | Color inkjet printer with wireless capability              |  142.99 |
|         8 | Camera              | Digital camera with 20MP resolution and 4K video recording |  879.99 |
|         9 | External Hard Drive | 2TB portable external hard drive                           |   87.99 |
|        10 | Gaming Console      | Latest gaming console with 1TB storage                     |  549.99 |
+-----------+---------------------+------------------------------------------------------------+---------+
10 rows in set (0.00 sec)

mysql>
```

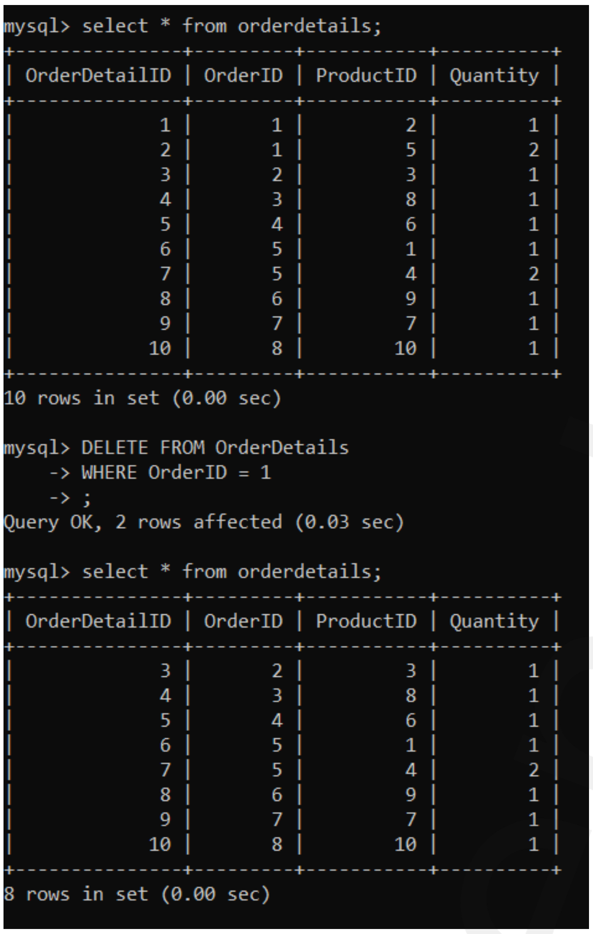
5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.

Ans.

-- Delete associated records from OrderDetails

DELETE FROM OrderDetails

WHERE OrderID = 1;



```
mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|             1 |       1 |         2 |        1 |
|             2 |       1 |         5 |        2 |
|             3 |       2 |         3 |        1 |
|             4 |       3 |         8 |        1 |
|             5 |       4 |         6 |        1 |
|             6 |       5 |         1 |        1 |
|             7 |       5 |         4 |        2 |
|             8 |       6 |         9 |        1 |
|             9 |       7 |         7 |        1 |
|            10 |       8 |        10 |        1 |
+---------------+---------+-----------+----------+
10 rows in set (0.00 sec)

mysql> DELETE FROM OrderDetails
    -> WHERE OrderID = 1
    -> ;
Query OK, 2 rows affected (0.03 sec)

mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|             3 |       2 |         3 |        1 |
|             4 |       3 |         8 |        1 |
|             5 |       4 |         6 |        1 |
|             6 |       5 |         1 |        1 |
|             7 |       5 |         4 |        2 |
|             8 |       6 |         9 |        1 |
|             9 |       7 |         7 |        1 |
|            10 |       8 |        10 |        1 |
+---------------+---------+-----------+----------+
8 rows in set (0.00 sec)
```

-- Delete the specific order from Orders

DELETE FROM Orders

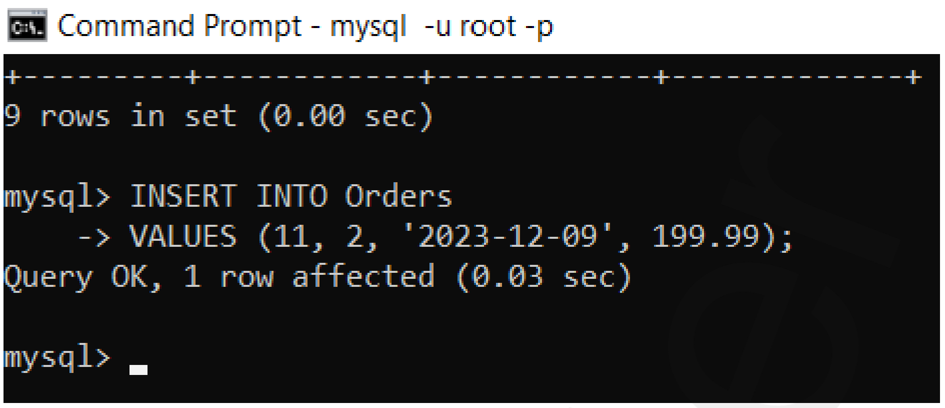
WHERE OrderID = @OrderIDToDelete;

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

Ans.

INSERT INTO Orders

VALUES (11, 2, '2023-12-09', 199.99);



Command Prompt - mysql -u root -p

```
+---------+------------+------------+-------------+
9 rows in set (0.00 sec)

mysql> INSERT INTO Orders
    -> VALUES (11, 2, '2023-12-09', 199.99);
Query OK, 1 row affected (0.03 sec)

mysql> _
```

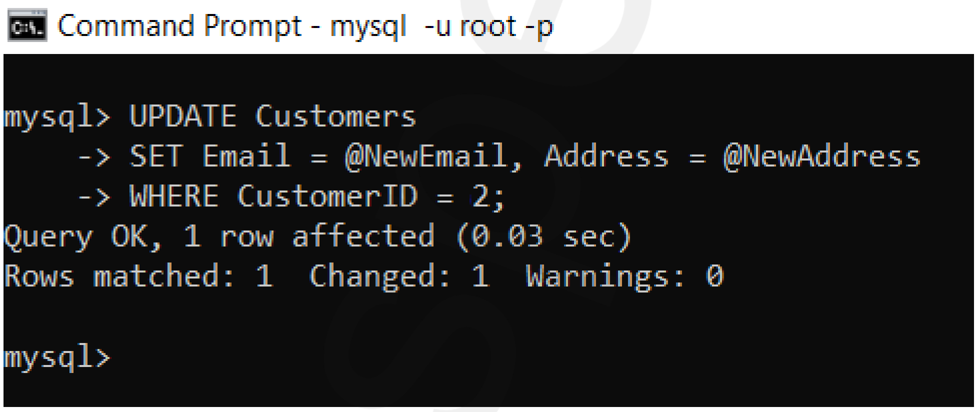
7. Write an SQL query to update the contact information (eg, email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information

Ans.

UPDATE Customers

SET Email = @NewEmail, Address = @NewAddress

WHERE CustomerID = 2;



Command Prompt - mysql -u root -p

```
mysql> UPDATE Customers
    -> SET Email = @NewEmail, Address = @NewAddress
    -> WHERE CustomerID = 2;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

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

Ans.

UPDATE Orders

SET TotalAmount = (

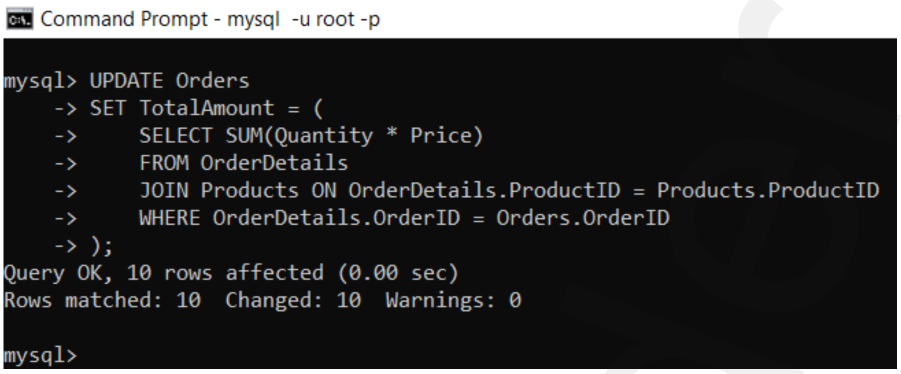
    SELECT SUM(Quantity * Price)

    FROM OrderDetails

    JOIN Products ON OrderDetails.ProductID = Products.ProductID

    WHERE OrderDetails.OrderID = Orders.OrderID

);

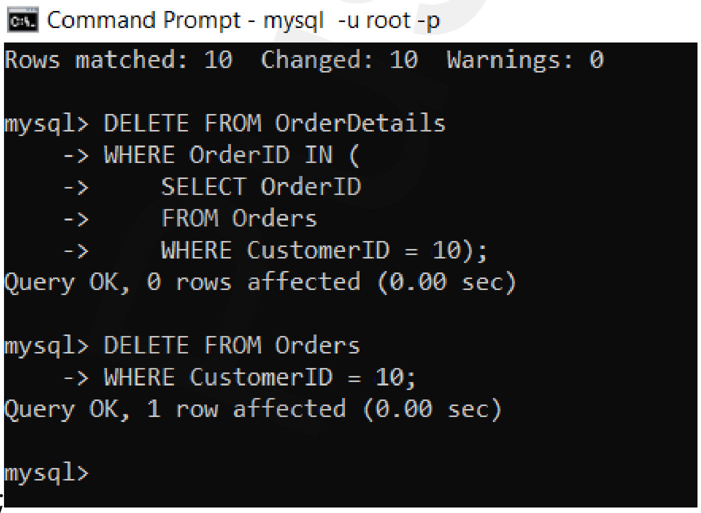


```
mysql> UPDATE Orders
    -> SET TotalAmount = (
    ->     SELECT SUM(Quantity * Price)
    ->     FROM OrderDetails
    ->     JOIN Products ON OrderDetails.ProductID = Products.ProductID
    ->     WHERE OrderDetails.OrderID = Orders.OrderID
    -> );
Query OK, 10 rows affected (0.00 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql>
```

9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OnderDetails" tables. Allow users to input the customer 10 as a parameter,

Ans.

```
DELETE FROM OrderDetails

WHERE OrderID IN (

    SELECT OrderID

    FROM Orders

    WHERE CustomerID = @CustomerIDToDelete


DELETE FROM Orders

WHERE CustomerID = @CustomerIDToDelete;
```



```
Rows matched: 10  Changed: 10  Warnings: 0

mysql> DELETE FROM OrderDetails
    -> WHERE OrderID IN (
    ->     SELECT OrderID
    ->     FROM Orders
    ->     WHERE CustomerID = 10);
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM Orders
    -> WHERE CustomerID = 10;
Query OK, 1 row affected (0.00 sec)

mysql>
```

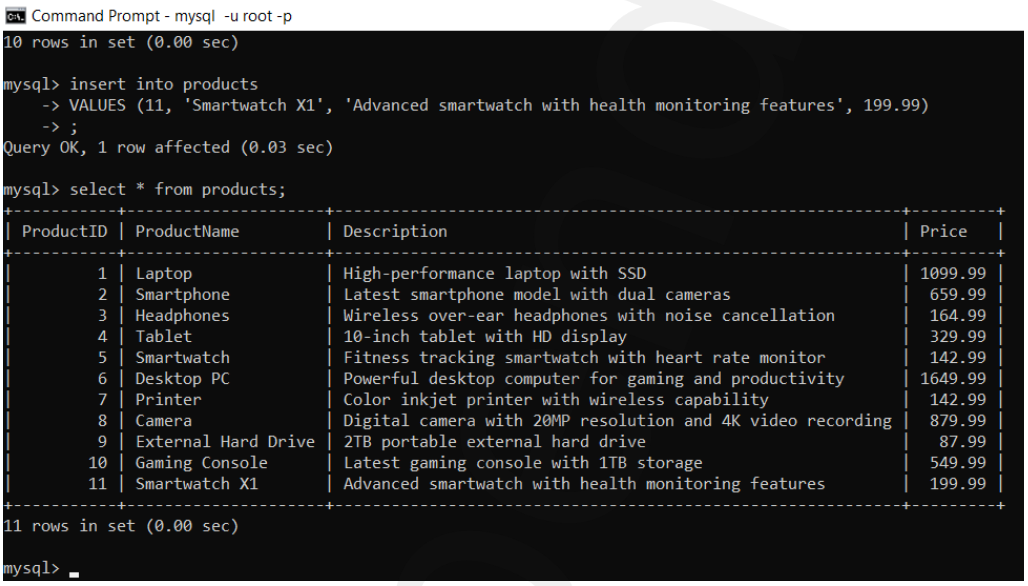
);

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

Ans.

INSERT INTO Products (ProductName, Description, Price)

VALUES ('Smartwatch X1', 'Advanced smartwatch with health monitoring features', 199.99);



```
10 rows in set (0.00 sec)

mysql> insert into products
    -> VALUES (11, 'Smartwatch X1', 'Advanced smartwatch with health monitoring features', 199.99)
    -> ;
Query OK, 1 row affected (0.03 sec)

mysql> select * from products;
+-----------+--------------------+----------------------------------------------------------+---------+
| ProductID | ProductName        | Description                                              | Price   |
+-----------+--------------------+----------------------------------------------------------+---------+
|         1 | Laptop             | High-performance laptop with SSD                         | 1099.99 |
|         2 | Smartphone         | Latest smartphone model with dual cameras                |  659.99 |
|         3 | Headphones         | Wireless over-ear headphones with noise cancellation     |  164.99 |
|         4 | Tablet             | 10-inch tablet with HD display                           |  329.99 |
|         5 | Smartwatch         | Fitness tracking smartwatch with heart rate monitor      |  142.99 |
|         6 | Desktop PC         | Powerful desktop computer for gaming and productivity    | 1649.99 |
|         7 | Printer            | Color inkjet printer with wireless capability            |  142.99 |
|         8 | Camera             | Digital camera with 20MP resolution and 4K video recording |  879.99 |
|         9 | External Hard Drive | 2TB portable external hard drive                        |   87.99 |
|        10 | Gaming Console     | Latest gaming console with 1TB storage                   |  549.99 |
|        11 | Smartwatch X1      | Advanced smartwatch with health monitoring features      |  199.99 |
+-----------+--------------------+----------------------------------------------------------+---------+
11 rows in set (0.00 sec)

mysql> _
```

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.

Ans.

-- Retrieve the number of orders placed by each customer

SELECT

    Customers.CustomerID,

    Customers.FirstName,

    Customers.LastName,

    COUNT(Orders.OrderID) AS NumberOfOrders

FROM

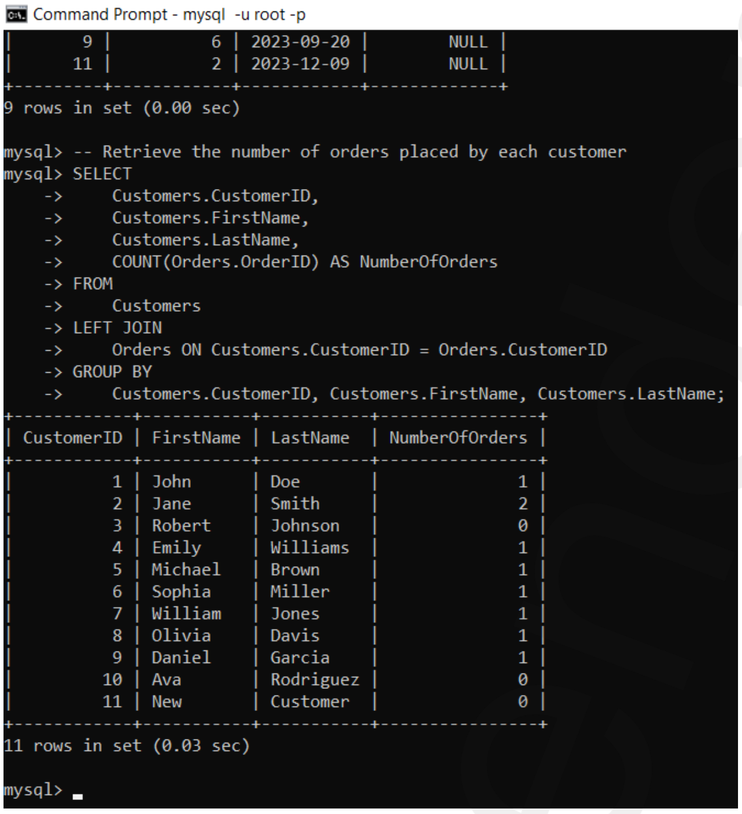
    Customers

LEFT JOIN

    Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY

    Customers.CustomerID, Customers.FirstName, Customers.LastName;


Command Prompt - mysql -u root -p

```
|       9 |           6 | 2023-09-20 |        NULL |
|      11 |           2 | 2023-12-09 |        NULL |
+---------+-------------+------------+-------------+
9 rows in set (0.00 sec)

mysql> -- Retrieve the number of orders placed by each customer
mysql> SELECT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     COUNT(Orders.OrderID) AS NumberOfOrders
    -> FROM
    ->     Customers
    -> LEFT JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName;
+------------+-----------+-----------+----------------+
| CustomerID | FirstName | LastName  | NumberOfOrders |
+------------+-----------+-----------+----------------+
|          1 | John      | Doe       |              1 |
|          2 | Jane      | Smith     |              2 |
|          3 | Robert    | Johnson   |              0 |
|          4 | Emily     | Williams  |              1 |
|          5 | Michael   | Brown     |              1 |
|          6 | Sophia    | Miller    |              1 |
|          7 | William   | Jones     |              1 |
|          8 | Olivia    | Davis     |              1 |
|          9 | Daniel    | Garcia    |              1 |
|         10 | Ava       | Rodriguez |              0 |
|         11 | New       | Customer  |              0 |
+------------+-----------+-----------+----------------+
11 rows in set (0.03 sec)

mysql> _
```

Task 3. Aggregate functions, Having, Order By, GroupBy 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.

Ans. SELECT

    Orders.OrderID,

    Customers.FirstName,

    Customers.LastName,

    Orders.OrderDate,

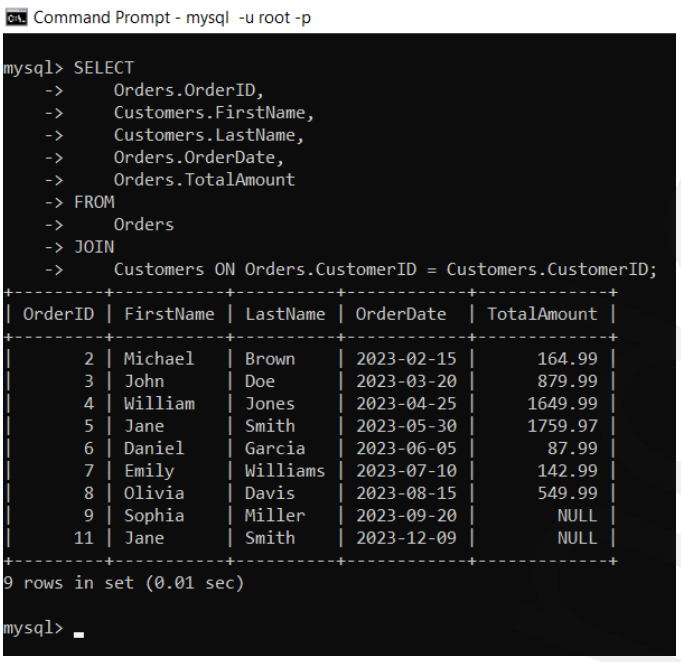
    Orders.TotalAmount

FROM

    Orders

JOIN

    Customers ON Orders.CustomerID = Customers.CustomerID;



```
Command Prompt - mysql  -u root -p

mysql> SELECT
    ->     Orders.OrderID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     Orders.OrderDate,
    ->     Orders.TotalAmount
    -> FROM
    ->     Orders
    -> JOIN
    ->     Customers ON Orders.CustomerID = Customers.CustomerID;
+---------+-----------+----------+------------+-------------+
| OrderID | FirstName | LastName | OrderDate  | TotalAmount |
+---------+-----------+----------+------------+-------------+
|       2 | Michael   | Brown    | 2023-02-15 |      164.99 |
|       3 | John      | Doe      | 2023-03-20 |      879.99 |
|       4 | William   | Jones    | 2023-04-25 |     1649.99 |
|       5 | Jane      | Smith    | 2023-05-30 |     1759.97 |
|       6 | Daniel    | Garcia   | 2023-06-05 |       87.99 |
|       7 | Emily     | Williams | 2023-07-10 |      142.99 |
|       8 | Olivia    | Davis    | 2023-08-15 |      549.99 |
|       9 | Sophia    | Miller   | 2023-09-20 |        NULL |
|      11 | Jane      | Smith    | 2023-12-09 |        NULL |
+---------+-----------+----------+------------+-------------+
9 rows in set (0.01 sec)

mysql>
```

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

Ans.

SELECT

    Products.ProductName,

    SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue

FROM

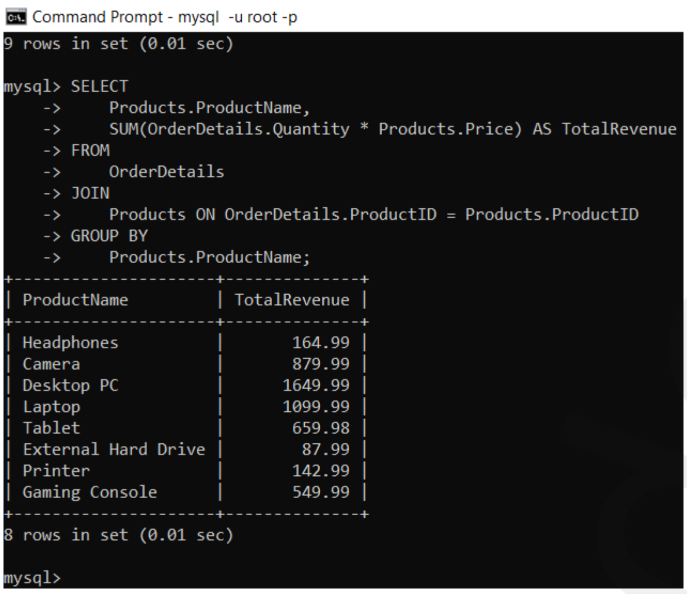
    OrderDetails

JOIN

    Products ON OrderDetails.ProductID = Products.ProductID

GROUP BY

    Products.ProductName;

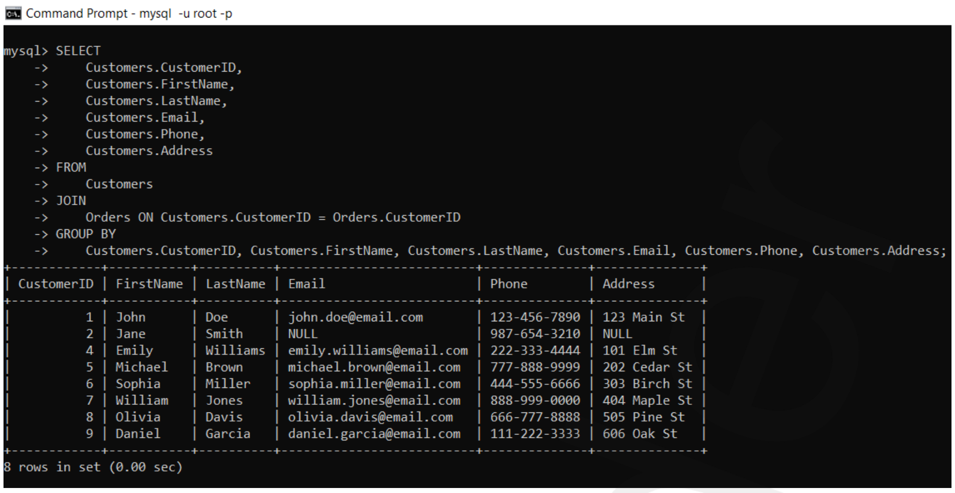


```
9 rows in set (0.01 sec)

mysql> SELECT
    ->     Products.ProductName,
    ->     SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue
    -> FROM
    ->     OrderDetails
    -> JOIN
    ->     Products ON OrderDetails.ProductID = Products.ProductID
    -> GROUP BY
    ->     Products.ProductName;
+--------------------+--------------+
| ProductName        | TotalRevenue |
+--------------------+--------------+
| Headphones         |       164.99 |
| Camera             |       879.99 |
| Desktop PC         |      1649.99 |
| Laptop             |      1099.99 |
| Tablet             |       659.98 |
| External Hard Drive |       87.99 |
| Printer            |       142.99 |
| Gaming Console     |       549.99 |
+--------------------+--------------+
8 rows in set (0.01 sec)

mysql>
```

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

Ans.

```
SELECT
    Customers.CustomerID,
    Customers.FirstName,
    Customers.LastName,
    Customers.Email,
    Customers.Phone,
    Customers.Address
FROM
    Customers
JOIN
    Orders ON Customers.CustomerID = Orders.CustomerID
GROUP BY
    Customers.CustomerID, Customers.FirstName, Customers.LastName, Customers.Email,
Customers.Phone, Customers.Address;
```



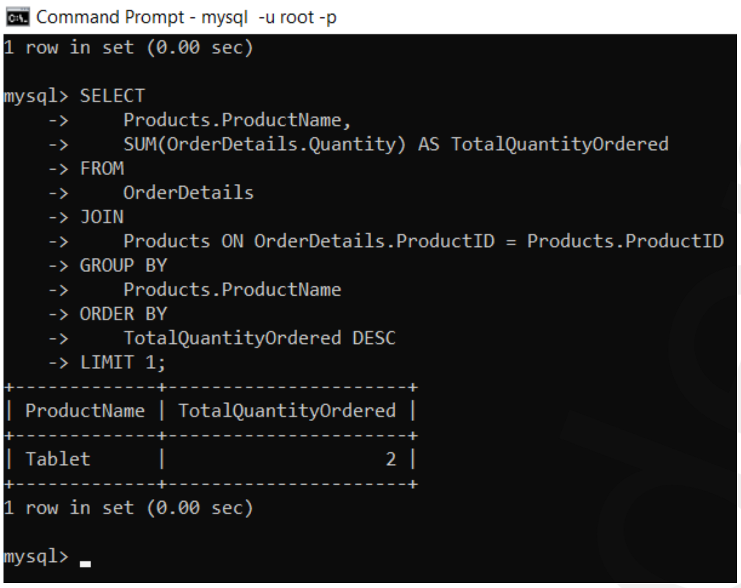
Command Prompt - mysql -u root -p

```
mysql> SELECT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     Customers.Email,
    ->     Customers.Phone,
    ->     Customers.Address
    -> FROM
    ->     Customers
    -> JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName, Customers.Email, Customers.Phone, Customers.Address;
+------------+-----------+----------+--------------------------+--------------+--------------+
| CustomerID | FirstName | LastName | Email                    | Phone        | Address      |
+------------+-----------+----------+--------------------------+--------------+--------------+
|          1 | John      | Doe      | john.doe@email.com       | 123-456-7890 | 123 Main St  |
|          2 | Jane      | Smith    | NULL                     | 987-654-3210 | NULL         |
|          4 | Emily     | Williams | emily.williams@email.com | 222-333-4444 | 101 Elm St   |
|          5 | Michael   | Brown    | michael.brown@email.com  | 777-888-9999 | 202 Cedar St |
|          6 | Sophia    | Miller   | sophia.miller@email.com  | 444-555-6666 | 303 Birch St |
|          7 | William   | Jones    | william.jones@email.com  | 888-999-0000 | 404 Maple St |
|          8 | Olivia    | Davis    | olivia.davis@email.com   | 666-777-8888 | 505 Pine St  |
|          9 | Daniel    | Garcia   | daniel.garcia@email.com  | 111-222-3333 | 606 Oak St   |
+------------+-----------+----------+--------------------------+--------------+--------------+
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.

Ans.

```
SELECT
    Products.ProductName,
    SUM(OrderDetails.Quantity) AS TotalQuantityOrdered
FROM
    OrderDetails
JOIN
    Products ON OrderDetails.ProductID = Products.ProductID
GROUP BY
    Products.ProductName
ORDER BY
    TotalQuantityOrdered DESC
LIMIT 1;
```



```
1 row in set (0.00 sec)

mysql> SELECT
    ->     Products.ProductName,
    ->     SUM(OrderDetails.Quantity) AS TotalQuantityOrdered
    -> FROM
    ->     OrderDetails
    -> JOIN
    ->     Products ON OrderDetails.ProductID = Products.ProductID
    -> GROUP BY
    ->     Products.ProductName
    -> ORDER BY
    ->     TotalQuantityOrdered DESC
    -> LIMIT 1;
+-------------+----------------------+
| ProductName | TotalQuantityOrdered |
+-------------+----------------------+
| Tablet      |                    2 |
+-------------+----------------------+
1 row in set (0.00 sec)

mysql>
```

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

Ans.

```
SELECT
    Products.ProductName,
    Products.Description,
    Products.Price,
    Categories.CategoryName
FROM
    Products
JOIN
    Categories ON Products.CategoryID = Categories.CategoryID
WHERE
    Categories.CategoryName = 'Electronic Gadgets';
```

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

Ans.

```
SELECT
```

Customers.CustomerID,

Customers.FirstName,

Customers.LastName,

AVG(Orders.TotalAmount) AS AverageOrderValue

FROM

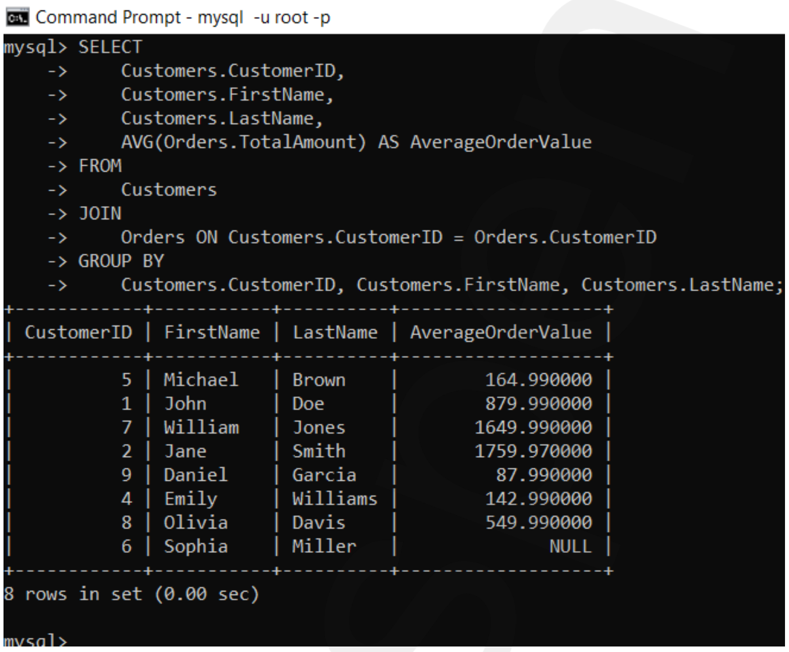
Customers

JOIN

Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY

Customers.CustomerID, Customers.FirstName, Customers.LastName;



```
mysql> SELECT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     AVG(Orders.TotalAmount) AS AverageOrderValue
    -> FROM
    ->     Customers
    -> JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName;
+------------+-----------+----------+-------------------+
| CustomerID | FirstName | LastName | AverageOrderValue |
+------------+-----------+----------+-------------------+
|          5 | Michael   | Brown    |        164.990000 |
|          1 | John      | Doe      |        879.990000 |
|          7 | William   | Jones    |       1649.990000 |
|          2 | Jane      | Smith    |       1759.970000 |
|          9 | Daniel    | Garcia   |         87.990000 |
|          4 | Emily     | Williams |        142.990000 |
|          8 | Olivia    | Davis    |        549.990000 |
|          6 | Sophia    | Miller   |              NULL |
+------------+-----------+----------+-------------------+
8 rows in set (0.00 sec)

mysql>
```

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

Ans.

SELECT

Orders.OrderID,

Customers.FirstName,

Customers.LastName,

Customers.Email,

Customers.Phone,

Customers.Address,

Orders.TotalAmount AS TotalRevenue

FROM

Orders

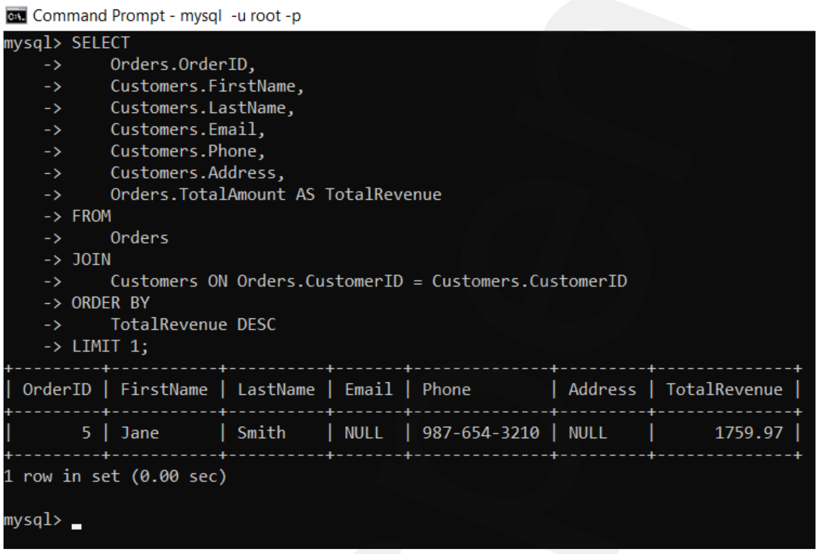
JOIN

Customers ON Orders.CustomerID = Customers.CustomerID

ORDER BY

TotalRevenue DESC

LIMIT 1;



Command Prompt - mysql -u root -p

```
mysql> SELECT
    ->     Orders.OrderID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     Customers.Email,
    ->     Customers.Phone,
    ->     Customers.Address,
    ->     Orders.TotalAmount AS TotalRevenue
    -> FROM
    ->     Orders
    -> JOIN
    ->     Customers ON Orders.CustomerID = Customers.CustomerID
    -> ORDER BY
    ->     TotalRevenue DESC
    -> LIMIT 1;
+---------+-----------+----------+-------+--------------+---------+--------------+
| OrderID | FirstName | LastName | Email | Phone        | Address | TotalRevenue |
+---------+-----------+----------+-------+--------------+---------+--------------+
|       5 | Jane      | Smith    | NULL  | 987-654-3210 | NULL    |      1759.97 |
+---------+-----------+----------+-------+--------------+---------+--------------+
1 row in set (0.00 sec)

mysql>
```

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

Ans.

SELECT

Products.ProductID,

Products.ProductName,

COUNT(OrderDetails.OrderID) AS NumberOfOrders

FROM

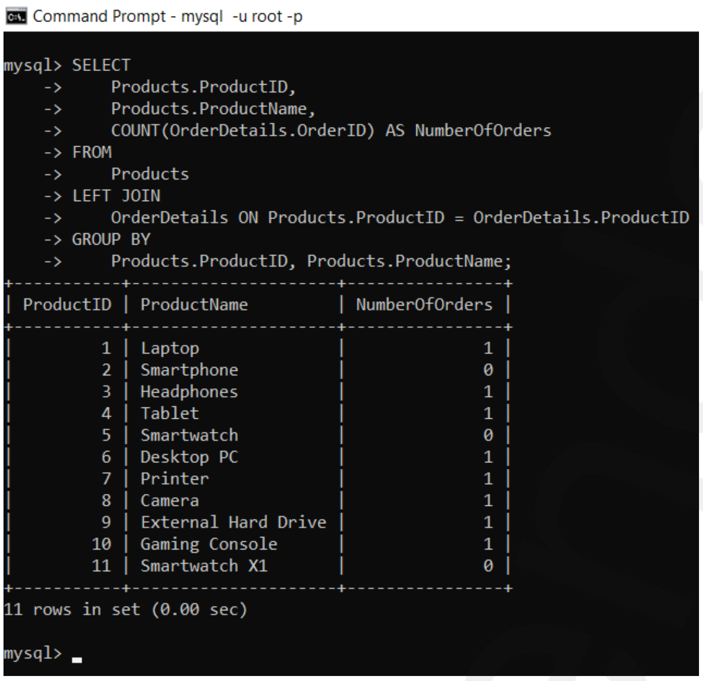
Products

LEFT JOIN

OrderDetails ON Products.ProductID = OrderDetails.ProductID

GROUP BY

Products.ProductID, Products.ProductName;



```
mysql> SELECT
    ->     Products.ProductID,
    ->     Products.ProductName,
    ->     COUNT(OrderDetails.OrderID) AS NumberOfOrders
    -> FROM
    ->     Products
    -> LEFT JOIN
    ->     OrderDetails ON Products.ProductID = OrderDetails.ProductID
    -> GROUP BY
    ->     Products.ProductID, Products.ProductName;
+-----------+---------------------+----------------+
| ProductID | ProductName         | NumberOfOrders |
+-----------+---------------------+----------------+
|         1 | Laptop              |              1 |
|         2 | Smartphone          |              0 |
|         3 | Headphones          |              1 |
|         4 | Tablet              |              1 |
|         5 | Smartwatch          |              0 |
|         6 | Desktop PC          |              1 |
|         7 | Printer             |              1 |
|         8 | Camera              |              1 |
|         9 | External Hard Drive |              1 |
|        10 | Gaming Console      |              1 |
|        11 | Smartwatch X1       |              0 |
+-----------+---------------------+----------------+
11 rows in set (0.00 sec)

mysql> _
```

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

Ans.

SELECT

Customers.CustomerID,

Customers.FirstName,

Customers.LastName,

Customers.Email,

Customers.Phone,

Customers.Address

FROM

Customers

JOIN

Orders ON Customers.CustomerID = Orders.CustomerID

JOIN

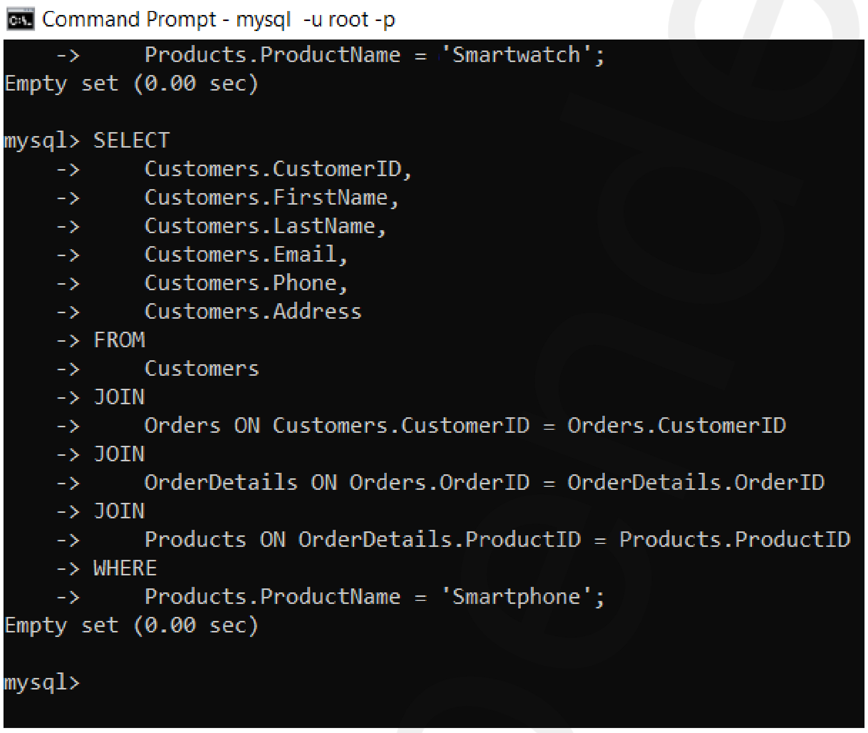
OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN

Products ON OrderDetails.ProductID = Products.ProductID

WHERE

Products.ProductName = 'Smartphone';


Command Prompt - mysql -u root -p

```
    ->     Products.ProductName = 'Smartwatch';
Empty set (0.00 sec)

mysql> SELECT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     Customers.Email,
    ->     Customers.Phone,
    ->     Customers.Address
    -> FROM
    ->     Customers
    -> JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN
    ->     OrderDetails ON Orders.OrderID = OrderDetails.OrderID
    -> JOIN
    ->     Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE
    ->     Products.ProductName = 'Smartphone';
Empty set (0.00 sec)

mysql>
```

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.

Ans.

SELECT

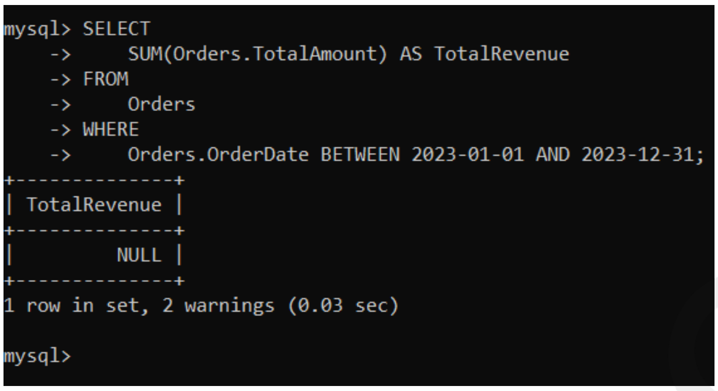
SUM(Orders.TotalAmount) AS TotalRevenue

FROM

Orders

WHERE

Orders.OrderDate BETWEEN 2023-01-01 AND 2023-12-31;


```
mysql> SELECT
    ->     SUM(Orders.TotalAmount) AS TotalRevenue
    -> FROM
    ->     Orders
    -> WHERE
    ->     Orders.OrderDate BETWEEN 2023-01-01 AND 2023-12-31;
+--------------+
| TotalRevenue |
+--------------+
|         NULL |
+--------------+
1 row in set, 2 warnings (0.03 sec)

mysql>
```

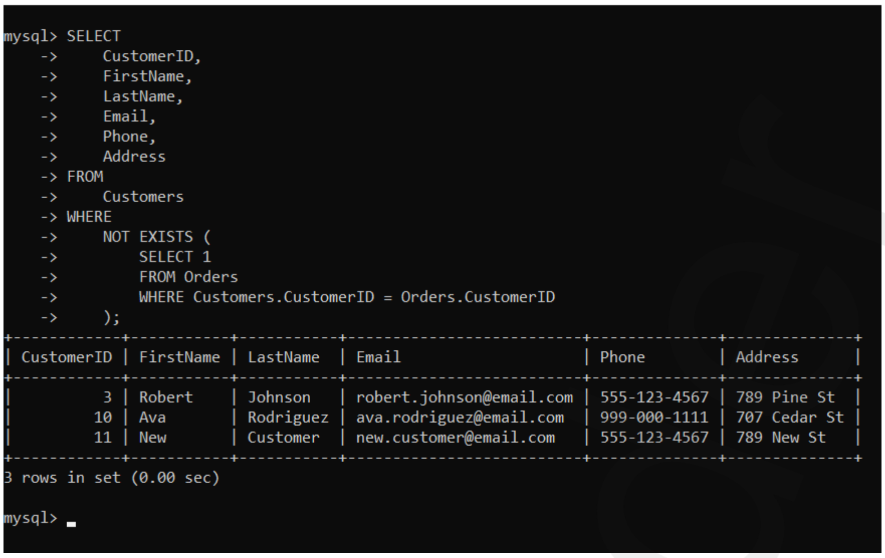
Task 4. Subquery and its type:

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

Ans.

```
SELECT
   CustomerID,
   FirstName,
   LastName,
   Email,
   Phone,
   Address
FROM
   Customers
WHERE
   NOT EXISTS (
      SELECT 1
      FROM Orders
      WHERE Customers.CustomerID = Orders.CustomerID
   );
```

Command Prompt - mysql -u root -p



```
mysql> SELECT
    ->     CustomerID,
    ->     FirstName,
    ->     LastName,
    ->     Email,
    ->     Phone,
    ->     Address
    -> FROM
    ->     Customers
    -> WHERE
    ->     NOT EXISTS (
    ->         SELECT 1
    ->         FROM Orders
    ->         WHERE Customers.CustomerID = Orders.CustomerID
    ->     );
+------------+-----------+-----------+---------------------------+--------------+--------------+
| CustomerID | FirstName | LastName  | Email                     | Phone        | Address      |
+------------+-----------+-----------+---------------------------+--------------+--------------+
|          3 | Robert    | Johnson   | robert.johnson@email.com  | 555-123-4567 | 789 Pine St  |
|         10 | Ava       | Rodriguez | ava.rodriguez@email.com   | 999-000-1111 | 707 Cedar St |
|         11 | New       | Customer  | new.customer@email.com    | 555-123-4567 | 789 New St   |
+------------+-----------+-----------+---------------------------+--------------+--------------+
3 rows in set (0.00 sec)

mysql> _
```

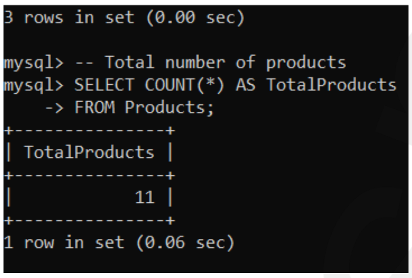
2. Write an SQL query to find number of products available for sale. generated by TechShop.

Ans.

SELECT COUNT(*) AS TotalProducts

FROM Products;

Command Prompt - mysql -u root -p



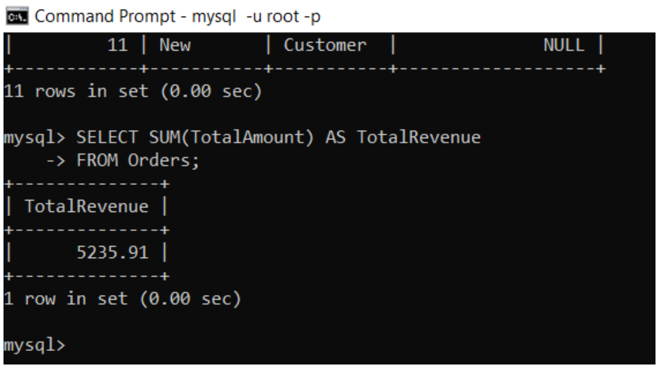
```
3 rows in set (0.00 sec)

mysql> -- Total number of products
mysql> SELECT COUNT(*) AS TotalProducts
    -> FROM Products;
+---------------+
| TotalProducts |
+---------------+
|            11 |
+---------------+
1 row in set (0.06 sec)
```

3. Write an SQL query to total revenue calculate the Allow users to input the category.

Ans.

SELECT SUM(TotalAmount) AS TotalRevenue

FROM Orders;



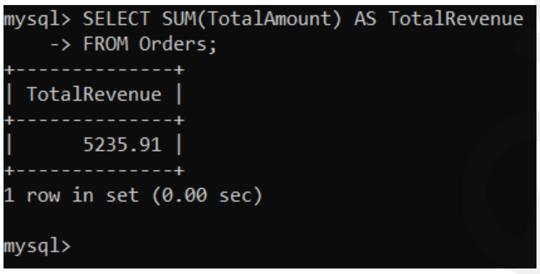
```
Command Prompt - mysql  -u root -p
|         11 | New        | Customer  |                NULL |
+------------+-----------+-----------+-------------------+
11 rows in set (0.00 sec)

mysql> SELECT SUM(TotalAmount) AS TotalRevenue
    -> FROM Orders;
+--------------+
| TotalRevenue |
+--------------+
|      5235.91 |
+--------------+
1 row in set (0.00 sec)

mysql>
```

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

Ans.

SELECT SUM(TotalAmount) AS TotalRevenue

FROM Orders;



```
mysql> SELECT SUM(TotalAmount) AS TotalRevenue
    -> FROM Orders;
+--------------+
| TotalRevenue |
+--------------+
|      5235.91 |
+--------------+
1 row in set (0.00 sec)

mysql>
```

5. 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.

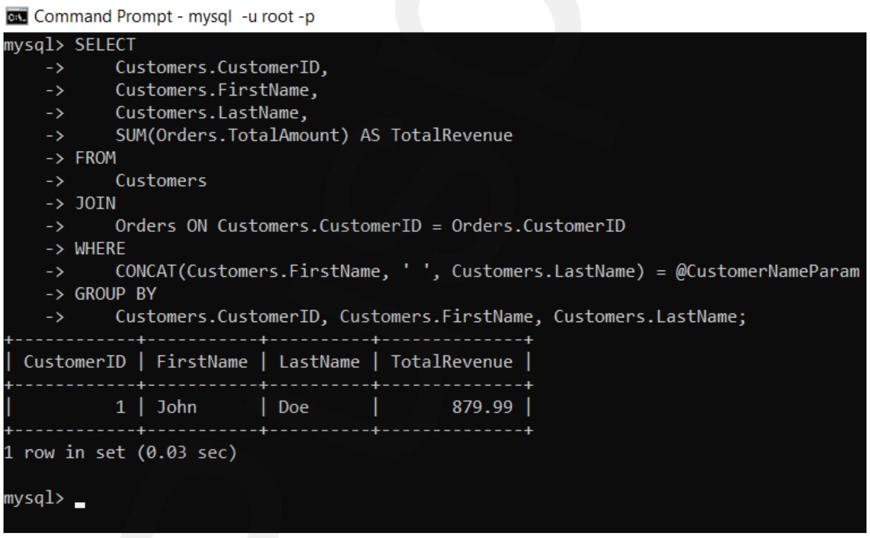
Ans.

```
SELECT
    AVG(OrderDetails.Quantity) AS AverageQuantityOrdered
FROM
    OrderDetails
JOIN
    Products ON OrderDetails.ProductID = Products.ProductID
JOIN
    Categories ON Products.CategoryID = Categories.CategoryID
WHERE
    Categories.CategoryName = @CategoryNameParam;
```

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

Ans.

```
SELECT

    Customers.CustomerID,

    Customers.FirstName,

    Customers.LastName,

    SUM(Orders.TotalAmount) AS TotalRevenue

FROM

    Customers

JOIN

    Orders ON Customers.CustomerID = Orders.CustomerID

WHERE

    CONCAT(Customers.FirstName, ' ', Customers.LastName) = 'John Doe'

GROUP BY

    Customers.CustomerID, Customers.FirstName, Customers.LastName;
```



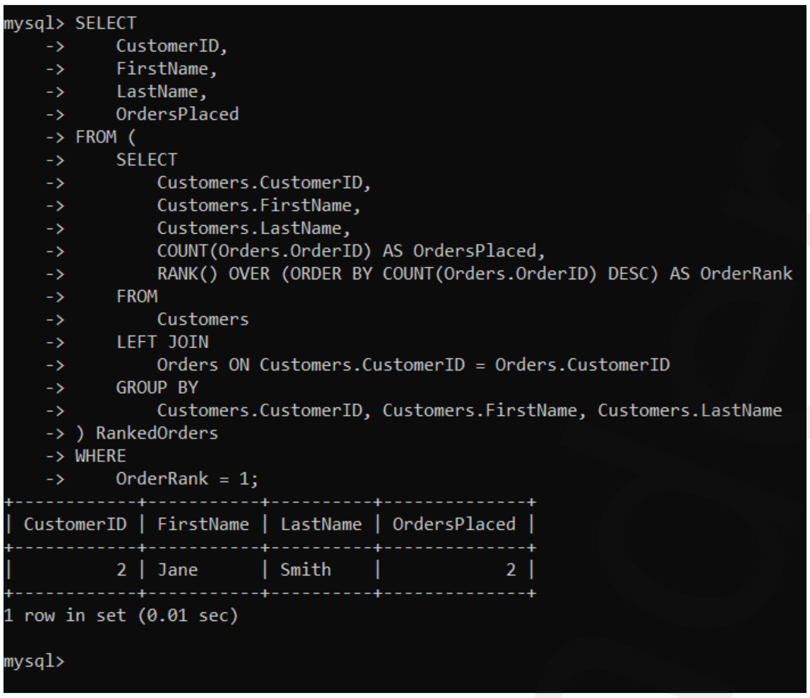
Command Prompt - mysql -u root -p

```
mysql> SELECT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     SUM(Orders.TotalAmount) AS TotalRevenue
    -> FROM
    ->     Customers
    -> JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> WHERE
    ->     CONCAT(Customers.FirstName, ' ', Customers.LastName) = @CustomerNameParam
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName;
+------------+-----------+----------+--------------+
| CustomerID | FirstName | LastName | TotalRevenue |
+------------+-----------+----------+--------------+
|          1 | John      | Doe      |       879.99 |
+------------+-----------+----------+--------------+
1 row in set (0.03 sec)

mysql> _
```

7. 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.

Ans.

```
SELECT
    CustomerID,
    FirstName,
    LastName,
    OrdersPlaced
FROM (
    SELECT
        Customers.CustomerID,
        Customers.FirstName,
        Customers.LastName,
        COUNT(Orders.OrderID) AS OrdersPlaced,
        RANK() OVER (ORDER BY COUNT(Orders.OrderID) DESC) AS OrderRank
    FROM
        Customers
    LEFT JOIN
        Orders ON Customers.CustomerID = Orders.CustomerID
    GROUP BY
        Customers.CustomerID, Customers.FirstName, Customers.LastName
) RankedOrders
WHERE
    OrderRank = 1;
```


```
mysql> SELECT
    ->     CustomerID,
    ->     FirstName,
    ->     LastName,
    ->     OrdersPlaced
    -> FROM (
    ->     SELECT
    ->         Customers.CustomerID,
    ->         Customers.FirstName,
    ->         Customers.LastName,
    ->         COUNT(Orders.OrderID) AS OrdersPlaced,
    ->         RANK() OVER (ORDER BY COUNT(Orders.OrderID) DESC) AS OrderRank
    ->     FROM
    ->         Customers
    ->     LEFT JOIN
    ->         Orders ON Customers.CustomerID = Orders.CustomerID
    ->     GROUP BY
    ->         Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> ) RankedOrders
    -> WHERE
    ->     OrderRank = 1;
+------------+-----------+----------+--------------+
| CustomerID | FirstName | LastName | OrdersPlaced |
+------------+-----------+----------+--------------+
|          2 | Jane      | Smith    |            2 |
+------------+-----------+----------+--------------+
1 row in set (0.01 sec)

mysql>
```

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

Ans.

```
SELECT
    CategoryName,
    TotalQuantityOrdered
FROM (
    SELECT
        Categories.CategoryName,
        SUM(OrderDetails.Quantity) AS TotalQuantityOrdered,
        RANK() OVER (ORDER BY SUM(OrderDetails.Quantity) DESC) AS CategoryRank
        Categories
    JOIN
        Products ON Categories.CategoryID = Products.CategoryID
    JOIN
        OrderDetails ON Products.ProductID = OrderDetails.ProductID
```

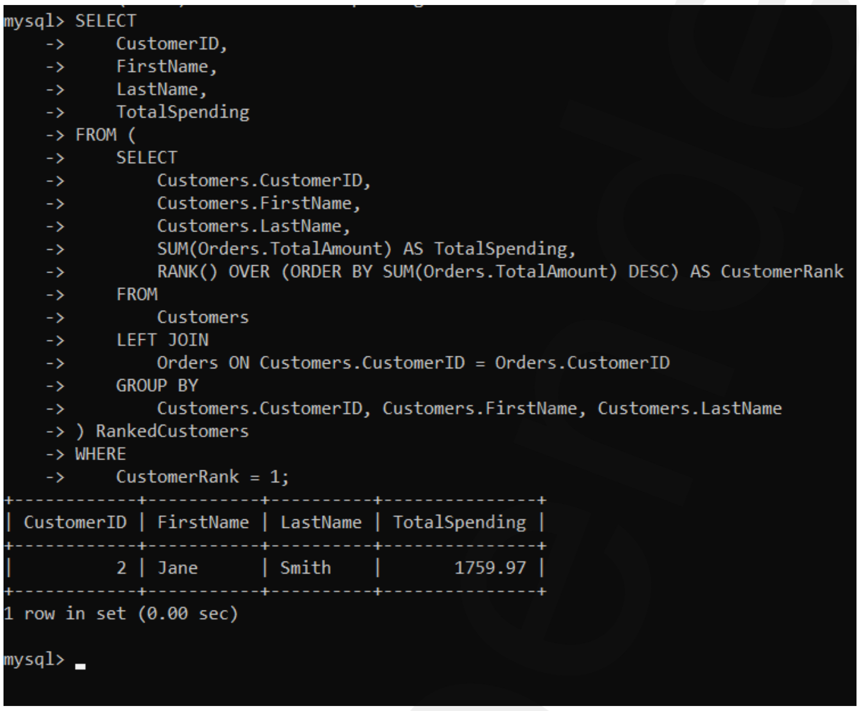
    GROUP BY

        Categories.CategoryName

) RankedCategories

WHERE

    CategoryRank = 1;



```
mysql> SELECT
    ->     CustomerID,
    ->     FirstName,
    ->     LastName,
    ->     TotalSpending
    -> FROM (
    ->     SELECT
    ->         Customers.CustomerID,
    ->         Customers.FirstName,
    ->         Customers.LastName,
    ->         SUM(Orders.TotalAmount) AS TotalSpending,
    ->         RANK() OVER (ORDER BY SUM(Orders.TotalAmount) DESC) AS CustomerRank
    ->     FROM
    ->         Customers
    ->     LEFT JOIN
    ->         Orders ON Customers.CustomerID = Orders.CustomerID
    ->     GROUP BY
    ->         Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> ) RankedCustomers
    -> WHERE
    ->     CustomerRank = 1;
+------------+-----------+----------+---------------+
| CustomerID | FirstName | LastName | TotalSpending |
+------------+-----------+----------+---------------+
|          2 | Jane      | Smith    |       1759.97 |
+------------+-----------+----------+---------------+
1 row in set (0.00 sec)

mysql> _
```

9. 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.

Ans.

SELECT

    Customers.CustomerID,

    Customers.FirstName,

    Customers.LastName,

    AVG(Orders.TotalAmount) AS AverageOrderValue

FROM

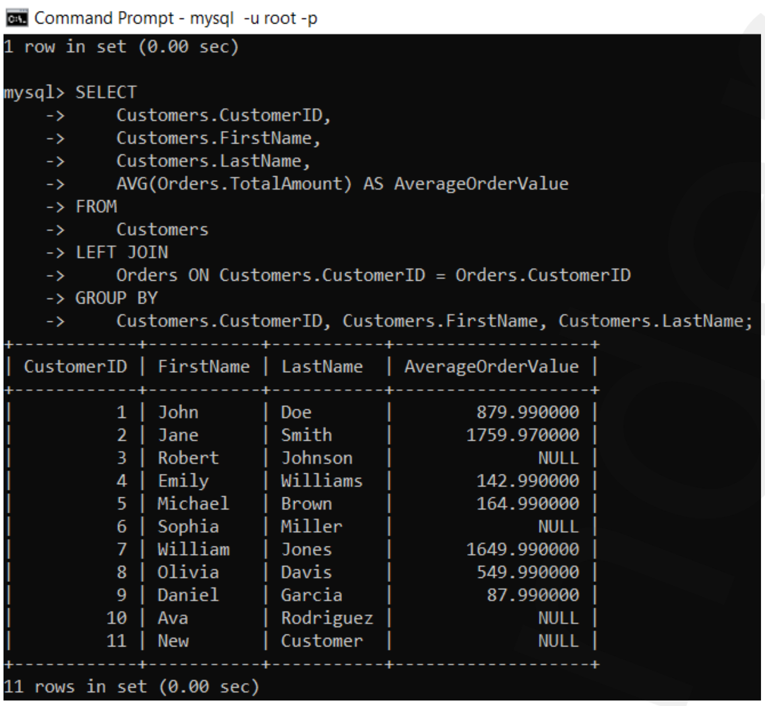
    Customers

LEFT JOIN

    Orders ON Customers.CustomerID = Orders.CustomerID

GROUP BY

Customers.CustomerID, Customers.FirstName, Customers.LastName;


Command Prompt - mysql -u root -p

```
1 row in set (0.00 sec)

mysql> SELECT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     AVG(Orders.TotalAmount) AS AverageOrderValue
    -> FROM
    ->     Customers
    -> LEFT JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName;
+------------+-----------+-----------+-------------------+
| CustomerID | FirstName | LastName  | AverageOrderValue |
+------------+-----------+-----------+-------------------+
|          1 | John      | Doe       |        879.990000 |
|          2 | Jane      | Smith     |       1759.970000 |
|          3 | Robert    | Johnson   |              NULL |
|          4 | Emily     | Williams  |        142.990000 |
|          5 | Michael   | Brown     |        164.990000 |
|          6 | Sophia    | Miller    |              NULL |
|          7 | William   | Jones     |       1649.990000 |
|          8 | Olivia    | Davis     |        549.990000 |
|          9 | Daniel    | Garcia    |         87.990000 |
|         10 | Ava       | Rodriguez |              NULL |
|         11 | New       | Customer  |              NULL |
+------------+-----------+-----------+-------------------+
11 rows in set (0.00 sec)
```