SQL Assignment1

Task1

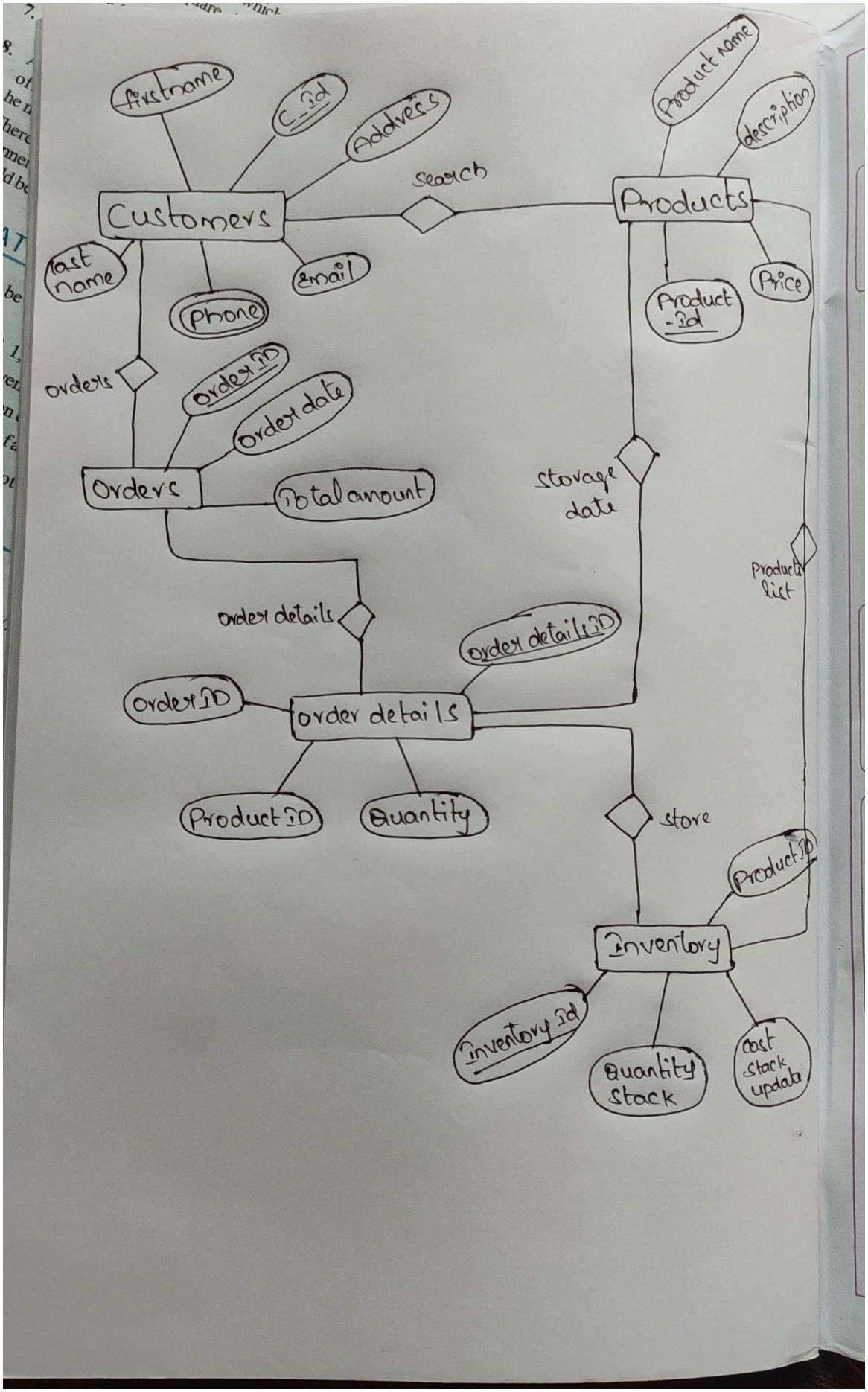
1. Create the database named "TechShop”

```
mysql> create database techshop;
Query OK, 1 row affected (0.01 sec)
```

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

```
mysql> CREATE TABLE Customers (CustomerID INT PRIMARY KEY,FirstName VARCHAR(50),LastName VARCHAR(50),Email VARCHAR(100),Phone VARCHAR(20),Addres
s VARCHAR(255));
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE Products (ProductID INT PRIMARY KEY,ProductName VARCHAR(100),Description TEXT,Price DECIMAL(10, 2));
Query OK, 0 rows affected (0.06 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 (0.10 sec)

mysql> CREATE TABLE OrderDetails (OrderDetailID INT PRIMARY KEY,OrderID INT,ProductID INT,Quantity INT,FOREIGN KEY (OrderID) REFERENCES Orders(O
rderID),FOREIGN KEY (ProductID) REFERENCES Products(ProductID));
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE Inventory (InventoryID INT PRIMARY KEY,ProductID INT,QuantityInStock INT,LastStockUpdate DATE,FOREIGN KEY (ProductID) REFERE
NCES Products(ProductID));
Query OK, 0 rows affected (0.07 sec)
```

3.Create an ERD for the database.


firstname
C_Id
Address
Product name
description
search
Customers
Products
last name
Email
Phone
Price
Product -Id
orders
order ID
order date
Orders
Total amount
storage date
Product list
order details
order details ID
order ID
order details
Product ID
Quantity
store
Product ID
Inventory
Inventory Id
Quantity stack
cost stack update

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

   b. Products

   c. Orders

   d. Order Details

   e. inventory

```
mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)VALUES(1, 'John', 'Doe', 'john.doe@example.com', '555-1234'
, '123 Main St'),(2, 'Jane', 'Smith', 'jane.smith@example.com', '555-5678', '456 Oak Ave'),(3, 'Bob', 'Johnson', 'bob.johnson@example.com', '555
-9876', '789 Pine Rd'),(4, 'Alice', 'Williams', 'alice.williams@example.com', '555-4321', '234 Cedar Ln'),(5, 'Charlie', 'Brown', 'charlie.brown
@example.com', '555-8765', '567 Maple Blvd'),(6, 'Eva', 'Davis', 'eva.davis@example.com', '555-2345', '890 Birch Dr'),(7, 'Frank', 'Miller', 'fr
ank.miller@example.com', '555-6789', '123 Pine Ave'),(8, 'Grace', 'Turner', 'grace.turner@example.com', '555-3456', '456 Elm St'),(9, 'David', '
Clark', 'david.clark@example.com', '555-7890', '789 Oak Ln'),(10, 'Helen', 'Ward', 'helen.ward@example.com', '555-0123', '234 Birch Rd');
Query OK, 10 rows affected (0.05 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Products (ProductID, ProductName, Description, Price)VALUES(1, 'Laptop', 'High-performance laptop with SSD', 999.99),(2, 'Sma
rtphone', 'Latest model with AI camera', 599.99),(3, 'Wireless Headphones', 'Noise-canceling wireless headphones', 149.99),(4, 'Smart Watch', 'F
itness tracking and smart notifications', 129.99),(5, '4K Smart TV', 'Ultra HD Smart TV with built-in streaming apps', 799.99),(6, 'Gaming Conso
le', 'Next-gen gaming console with 1TB storage', 499.99),(7, 'Digital Camera', 'Mirrorless camera with 24MP sensor', 799.99),(8, 'Bluetooth Spea
ker', 'Portable speaker with long battery life', 79.99),(9, 'External Hard Drive', '2TB USB 3.0 External HDD', 89.99),(10, 'Tablet', '10-inch ta
```

```
mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES(1,1, '2024-01-16', 150.00),(2,2, '2024-01-17', 200.50),(3,3, '202
4-01-18', 75.20),(4,4, '2024-01-19', 120.75),(5,5, '2024-01-20', 90.30),(6,6, '2024-01-21', 180.60),(7,7, '2024-01-22', 220.00),(8,8, '2024-01-2
3', 130.25),(9,9, '2024-01-24', 95.80),(10,10, '2024-01-25', 210.90);
Query OK, 10 rows affected (0.04 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)VALUES(1, 1, 1, 2),(2, 2, 2, 1),(3, 3, 3, 3),(4, 4, 4, 1),(5, 5, 5,
 2),(6, 6, 6, 2),(7, 7, 7, 1),(8, 8, 8, 3),(9, 9, 9, 1),(10,10, 10, 2);
Query OK, 10 rows affected (0.04 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)VALUES(1, 1, 50, '2024-01-16'),(2, 2, 30, '2024-01-15'),(
3, 3, 20, '2024-01-14'),(4, 4, 40, '2024-01-13'),(5, 5, 25, '2024-01-12'),(6, 6, 15, '2024-01-11'),(7, 7, 10, '2024-01-10'),(8, 8, 35, '2024-01-
09'),(9, 9, 18, '2024-01-08'),(10, 10, 22, '2024-01-07');
Query OK, 10 rows affected (0.04 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Task2:

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

```
mysql> SELECT FirstName, LastName, Email FROM Customers;
+-----------+----------+---------------------------+
| FirstName | LastName | Email                     |
+-----------+----------+---------------------------+
| John      | Doe      | john.doe@example.com      |
| Jane      | Smith    | jane.smith@example.com    |
| Bob       | Johnson  | bob.johnson@example.com   |
| Alice     | Williams | alice.williams@example.com |
| Charlie   | Brown    | charlie.brown@example.com |
| Eva       | Davis    | eva.davis@example.com     |
| Frank     | Miller   | frank.miller@example.com  |
| Grace     | Turner   | grace.turner@example.com  |
| David     | Clark    | david.clark@example.com   |
| Helen     | Ward     | helen.ward@example.com    |
+-----------+----------+---------------------------+
10 rows in set (0.00 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 JOIN Customers ON Orders.CustomerID = Custom
ers.CustomerID;
+---------+------------+-----------+----------+
| OrderID | OrderDate  | FirstName | LastName |
+---------+------------+-----------+----------+
|       1 | 2024-01-16 | John      | Doe      |
|       2 | 2024-01-17 | Jane      | Smith    |
|       3 | 2024-01-18 | Bob       | Johnson  |
|       4 | 2024-01-19 | Alice     | Williams |
|       5 | 2024-01-20 | Charlie   | Brown    |
|       6 | 2024-01-21 | Eva       | Davis    |
|       7 | 2024-01-22 | Frank     | Miller   |
|       8 | 2024-01-23 | Grace     | Turner   |
|       9 | 2024-01-24 | David     | Clark    |
|      10 | 2024-01-25 | Helen     | Ward     |
+---------+------------+-----------+----------+
10 rows in set (0.00 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> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)VALUES (11, 'John', 'Doe', 'john.doe@example.com', '123-456
-7890', '123 Main Street, Cityville');
Query OK, 1 row affected (0.04 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 * 1.10;
Query OK, 10 rows affected, 10 warnings (0.02 sec)
Rows matched: 10  Changed: 10  Warnings: 10
```

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> DELETE FROM OrderDetails WHERE OrderID = @OrderID;
Query OK, 0 rows affected (0.04 sec)

mysql> DELETE FROM OrderDetails WHERE OrderID = 10;
Query OK, 1 row affected (0.04 sec)

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

mysql> COMMIT;
Query OK, 0 rows 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> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES (10,10, '2024-01-17', 0.00);
Query OK, 1 row affected (0.04 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.

```
mysql> UPDATE Customers SET Email = @NewEmail,Address = @NewAddress WHERE CustomerID = 10;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

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(Quantity * Products.Price)FROM OrderDetails JOIN Products ON OrderDetails.ProductID = Product
s.ProductID WHERE Orders.OrderID = OrderDetails.OrderID)WHERE OrderID IN (SELECT OrderID FROM OrderDetails);
Query OK, 9 rows affected (0.01 sec)
Rows matched: 9  Changed: 9  Warnings: 0
```

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> DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = 10);
Query OK, 0 rows affected (0.04 sec)

mysql> DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = 9);
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM Orders WHERE CustomerID = 9;
Query OK, 1 row affected (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> INSERT INTO Products (ProductID, ProductName, Description, Price) VALUES (11,'TV', 'Description of the new gadget', 499.99);
Query OK, 1 row affected (0.04 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> UPDATE Orders SET OrderDate = '2024-01-01' WHERE OrderID = 1;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

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> UPDATE Customers SET NumberOfOrders = (SELECT COUNT(OrderID)FROM Orders WHERE Customers.CustomerID = Orders.CustomerID);
Query OK, 11 rows affected (0.04 sec)
Rows matched: 11  Changed: 11  Warnings: 0
```

Task3:

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

```
mysql> SELECT Orders.OrderID,Customers.FirstName,Customers.LastName,Orders.OrderDate,Orders.TotalAmount FROM Orders JOIN Customers ON Orders.CustomerID = Customers.Cust
erID;
+---------+-----------+----------+------------+-------------+
| OrderID | FirstName | LastName | OrderDate  | TotalAmount |
+---------+-----------+----------+------------+-------------+
|       1 | John      | Doe      | 2024-01-01 |     2199.98 |
|       2 | Jane      | Smith    | 2024-01-17 |      659.99 |
|       3 | Bob       | Johnson  | 2024-01-18 |      494.97 |
|       4 | Alice     | Williams | 2024-01-19 |      142.99 |
|       5 | Charlie   | Brown    | 2024-01-20 |     1759.98 |
|       6 | Eva       | Davis    | 2024-01-21 |     1099.98 |
|       7 | Frank     | Miller   | 2024-01-22 |      879.99 |
|       8 | Grace     | Turner   | 2024-01-23 |      263.97 |
|      10 | Helen     | Ward     | 2024-01-17 |        0.00 |
+---------+-----------+----------+------------+-------------+
9 rows in set (0.00 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> SELECT P.ProductID,P.ProductName,SUM(OD.Quantity * P.Price) AS TotalRevenue FROM Products AS P JOIN OrderDetails AS OD ON P.ProductID = OD.ProductID JOIN Orders
 O ON OD.OrderID = O.OrderID GROUP BY P.ProductID, P.ProductName;
+-----------+---------------------+--------------+
| ProductID | ProductName         | TotalRevenue |
+-----------+---------------------+--------------+
|         1 | Laptop              |      2199.98 |
|         2 | Smartphone          |       659.99 |
|         3 | Wireless Headphones |       494.97 |
|         4 | Smart Watch         |       142.99 |
|         5 | 4K Smart TV         |      1759.98 |
|         6 | Gaming Console      |      1099.98 |
|         7 | Digital Camera      |       879.99 |
|         8 | Bluetooth Speaker   |       263.97 |
+-----------+---------------------+--------------+
8 rows in set (0.00 sec)
```

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

```
mysql> SELECT Customers.CustomerID,Customers.FirstName,Customers.LastName,Customers.Email,Customers.Phone,Customers.Address FROM Customers WHERE EXISTS (SELECT 1 FROM Or
ers WHERE Orders.CustomerID = Customers.CustomerID);
+------------+-----------+----------+--------------------------+----------+----------------+
| CustomerID | FirstName | LastName | Email                    | Phone    | Address        |
+------------+-----------+----------+--------------------------+----------+----------------+
|          1 | John      | Doe      | john.doe@example.com     | 555-1234 | 123 Main St    |
|          2 | Jane      | Smith    | jane.smith@example.com   | 555-5678 | 456 Oak Ave    |
|          3 | Bob       | Johnson  | bob.johnson@example.com  | 555-9876 | 789 Pine Rd    |
|          4 | Alice     | Williams | alice.williams@example.com | 555-4321 | 234 Cedar Ln   |
|          5 | Charlie   | Brown    | charlie.brown@example.com | 555-8765 | 567 Maple Blvd |
|          6 | Eva       | Davis    | eva.davis@example.com    | 555-2345 | 890 Birch Dr   |
|          7 | Frank     | Miller   | frank.miller@example.com | 555-6789 | 123 Pine Ave   |
|          8 | Grace     | Turner   | grace.turner@example.com | 555-3456 | 456 Elm St     |
|         10 | Helen     | Ward     | NULL                     | 555-0123 | NULL           |
+------------+-----------+----------+--------------------------+----------+----------------+
9 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> SELECT P.ProductName,SUM(OD.Quantity) AS TotalQuantityOrdered FROM Products AS P JOIN OrderDetails AS OD ON P.ProductID = OD.ProductID GROUP BY P.ProductName ORD
 BY TotalQuantityOrdered DESC LIMIT 1;
+-----------------+----------------------+
| ProductName     | TotalQuantityOrdered |
+-----------------+----------------------+
| Bluetooth Speaker |                    3 |
+-----------------+----------------------+
1 row in set (0.00 sec)
```

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

```
mysql> select ProductID,ProductName from Products;
+-----------+---------------------+
| ProductID | ProductName         |
+-----------+---------------------+
|         1 | Laptop              |
|         2 | Smartphone          |
|         3 | Wireless Headphones |
|         4 | Smart Watch         |
|         5 | 4K Smart TV         |
|         6 | Gaming Console      |
|         7 | Digital Camera      |
|         8 | Bluetooth Speaker   |
|         9 | External Hard Drive |
|        10 | Tablet              |
|        11 | TV                  |
+-----------+---------------------+
11 rows in set (0.00 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> SELECT C.CustomerID,C.FirstName,C.LastName,AVG(O.TotalAmount) AS AverageOrderValue FROM Customers AS C JOIN Orders AS O ON C.CustomerID = O.CustomerID GROUP BY C.C
ustomerID, C.FirstName, C.LastName;
+------------+-----------+----------+--------------------+
| CustomerID | FirstName | LastName | AverageOrderValue |
+------------+-----------+----------+--------------------+
|          1 | John      | Doe      |       2199.980000 |
|          2 | Jane      | Smith    |        659.990000 |
|          3 | Bob       | Johnson  |        494.970000 |
|          4 | Alice     | Williams |        142.990000 |
|          5 | Charlie   | Brown    |       1759.980000 |
|          6 | Eva       | Davis    |       1099.980000 |
|          7 | Frank     | Miller   |        879.990000 |
|          8 | Grace     | Turner   |        263.970000 |
|         10 | Helen     | Ward     |          0.000000 |
+------------+-----------+----------+--------------------+
9 rows in set (0.01 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> SELECT O.OrderID,C.CustomerID,C.FirstName,C.LastName,O.TotalAmount AS TotalRevenue FROM Orders AS O JOIN Customers AS C ON O.CustomerID = C.CustomerID ORDER BY To
alRevenue DESC LIMIT 1;
+---------+------------+-----------+----------+--------------+
| OrderID | CustomerID | FirstName | LastName | TotalRevenue |
+---------+------------+-----------+----------+--------------+
|       1 |          1 | John      | Doe      |      2199.98 |
+---------+------------+-----------+----------+--------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT P.ProductID,P.ProductName,COUNT(OD.ProductID) AS NumberOfOrders FROM Products AS P LEFT JOIN OrderDetails AS OD ON P.ProductID = OD.ProductID GROUP BY P.Pr
ductID, P.ProductName ORDER BY NumberOfOrders DESC;
+-----------+---------------------+----------------+
| ProductID | ProductName         | NumberOfOrders |
+-----------+---------------------+----------------+
|         1 | Laptop              |              1 |
|         2 | Smartphone          |              1 |
|         3 | Wireless Headphones |              1 |
|         4 | Smart Watch         |              1 |
|         5 | 4K Smart TV         |              1 |
|         6 | Gaming Console      |              1 |
|         7 | Digital Camera      |              1 |
|         8 | Bluetooth Speaker   |              1 |
|         9 | External Hard Drive |              0 |
|        10 | Tablet              |              0 |
|        11 | TV                  |              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> SELECT C.CustomerID,ProductName,C.FirstName,C.LastName,C.Email,C.Phone,C.Address FROM Customers AS C JOIN Orders AS O ON C.CustomerID = O.CustomerID JOIN OrderDe
ils AS OD ON O.OrderID = OD.OrderID JOIN Products AS P ON OD.ProductID = P.ProductID;
+------------+---------------------+-----------+----------+---------------------------+----------+----------------+
| CustomerID | ProductName         | FirstName | LastName | Email                     | Phone    | Address        |
+------------+---------------------+-----------+----------+---------------------------+----------+----------------+
|          1 | Laptop              | John      | Doe      | john.doe@example.com      | 555-1234 | 123 Main St    |
|          2 | Smartphone          | Jane      | Smith    | jane.smith@example.com    | 555-5678 | 456 Oak Ave    |
|          3 | Wireless Headphones | Bob       | Johnson  | bob.johnson@example.com   | 555-9876 | 789 Pine Rd    |
|          4 | Smart Watch         | Alice     | Williams | alice.williams@example.com| 555-4321 | 234 Cedar Ln   |
|          5 | 4K Smart TV         | Charlie   | Brown    | charlie.brown@example.com | 555-8765 | 567 Maple Blvd |
|          6 | Gaming Console      | Eva       | Davis    | eva.davis@example.com     | 555-2345 | 890 Birch Dr   |
|          7 | Digital Camera      | Frank     | Miller   | frank.miller@example.com  | 555-6789 | 123 Pine Ave   |
|          8 | Bluetooth Speaker   | Grace     | Turner   | grace.turner@example.com  | 555-3456 | 456 Elm St     |
+------------+---------------------+-----------+----------+---------------------------+----------+----------------+
8 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> SELECT SUM(O.TotalAmount) AS TotalRevenue FROM Orders AS O WHERE O.OrderDate BETWEEN '2024-01-17' AND '2024-01-20';
+--------------+
| TotalRevenue |
+--------------+
|      3057.93 |
+--------------+
1 row in set (0.00 sec)
```

Task4:

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

```
mysql> SELECT C.CustomerID,C.FirstName,C.LastName,C.Email,C.Phone,C.Address FROM Customers AS C LEFT JOIN Orders AS O ON C.CustomerID = O.CustomerID WHERE O.OrderID IS NU
LL;
+------------+-----------+----------+-------------------+--------------+---------------------------+
| CustomerID | FirstName | LastName | Email             | Phone        | Address                   |
+------------+-----------+----------+-------------------+--------------+---------------------------+
|          9 | David     | Clark    | david.clark@example.com | 555-7890     | 789 Oak Ln                |
|         11 | John      | Doe      | john.doe@example.com    | 123-456-7890 | 123 Main Street, Cityville |
+------------+-----------+----------+-------------------+--------------+---------------------------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT COUNT(*) AS TotalProducts FROM Products;
+---------------+
| TotalProducts |
+---------------+
|            11 |
+---------------+
1 row in set (0.01 sec)
```

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

```
mysql> SELECT SUM(TotalAmount) AS TotalRevenue FROM Orders;
+--------------+
| TotalRevenue |
+--------------+
|      7501.85 |
+--------------+
1 row in set (0.00 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> SELECT AVG(OD.Quantity) AS AverageQuantityOrdered FROM OrderDetails AS OD JOIN Products AS P ON OD.ProductID = P.ProductID WHERE P.ProductName = 'Tablet';
+-----------------------+
| AverageQuantityOrdered |
+-----------------------+
|                  NULL |
+-----------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT AVG(OD.Quantity) AS AverageQuantityOrdered FROM OrderDetails AS OD JOIN Products AS P ON OD.ProductID = P.ProductID WHERE P.ProductName = 'Laptop';
+-----------------------+
| AverageQuantityOrdered |
+-----------------------+
|                2.0000 |
+-----------------------+
1 row in set (0.00 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> SELECT SUM(TotalAmount) AS TotalRevenue FROM Orders WHERE CustomerID = 3;
+--------------+
| TotalRevenue |
+--------------+
|       494.97 |
+--------------+
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> SELECT C.CustomerID,C.FirstName,C.LastName,COUNT(O.OrderID) AS NumberOfOrders FROM Customers AS C JOIN Orders AS O ON C.CustomerID = O.CustomerID GROUP BY C.Custom
erID, C.FirstName, C.LastName ORDER BY NumberOfOrders DESC LIMIT 1;
+------------+-----------+----------+----------------+
| CustomerID | FirstName | LastName | NumberOfOrders |
+------------+-----------+----------+----------------+
|          1 | John      | Doe      |              1 |
+------------+-----------+----------+----------------+
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> SELECT P.ProductName,SUM(OD.Quantity) AS TotalQuantityOrdered FROM OrderDetails AS OD JOIN Products AS P ON OD.ProductID = P.ProductID GROUP BY P.ProductName ORDER
 BY TotalQuantityOrdered DESC LIMIT 1;
+-----------------+----------------------+
| ProductName     | TotalQuantityOrdered |
+-----------------+----------------------+
| Bluetooth Speaker |                    3 |
+-----------------+----------------------+
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> SELECT C.CustomerID,C.FirstName,C.LastName,SUM(O.TotalAmount) AS TotalSpending FROM Customers AS C JOIN Orders AS O ON C.CustomerID = O.CustomerID JOIN OrderDetail
s AS OD ON O.OrderID = OD.OrderID JOIN Products AS P ON OD.ProductID = P.ProductID GROUP BY C.CustomerID, C.FirstName, C.LastName ORDER BY TotalSpending DESC LIMIT 1;
+------------+-----------+----------+--------------+
| CustomerID | FirstName | LastName | TotalSpending |
+------------+-----------+----------+--------------+
|          1 | John      | Doe      |      2199.98 |
+------------+-----------+----------+--------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT AVG(O.TotalAmount) AS AverageOrderValue FROM Orders AS O;
+--------------------+
| AverageOrderValue |
+--------------------+
|         833.538889 |
+--------------------+
1 row 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> SELECT C.CustomerID,C.FirstName,C.LastName,COUNT(O.OrderID) AS OrderCount FROM Customers AS C LEFT JOIN Orders AS O ON C.CustomerID = O.CustomerID GROUP BY C.Custo
merID, C.FirstName, C.LastName ORDER BY OrderCount DESC;
+------------+-----------+----------+------------+
| CustomerID | FirstName | LastName | OrderCount |
+------------+-----------+----------+------------+
|          1 | John      | Doe      |          1 |
|          2 | Jane      | Smith    |          1 |
|          3 | Bob       | Johnson  |          1 |
|          4 | Alice     | Williams |          1 |
|          5 | Charlie   | Brown    |          1 |
|          6 | Eva       | Davis    |          1 |
|          7 | Frank     | Miller   |          1 |
|          8 | Grace     | Turner   |          1 |
|         10 | Helen     | Ward     |          1 |
|          9 | David     | Clark    |          0 |
|         11 | John      | Doe      |          0 |
+------------+-----------+----------+------------+
11 rows in set (0.00 sec)
```