TechShop, an electronic gadgets shop

Task:1. Database Design:

1. Create the database named "TechShop"

```
mysql> CREATE DATABASE TechShop;
Query OK, 1 row affected (0.04 sec)

mysql> use TechShop;
```

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

```
mysql> CREATE TABLE Customers (
    -> CustomerID INT PRIMARY KEY,
    -> FirstName VARCHAR(225),
    -> LastName VARCHAR(225),
    -> Email TEXT,
    -> Phone BIGINT,
    -> Address TEXT
    -> );
Query OK, 0 rows affected (0.10 sec)
```

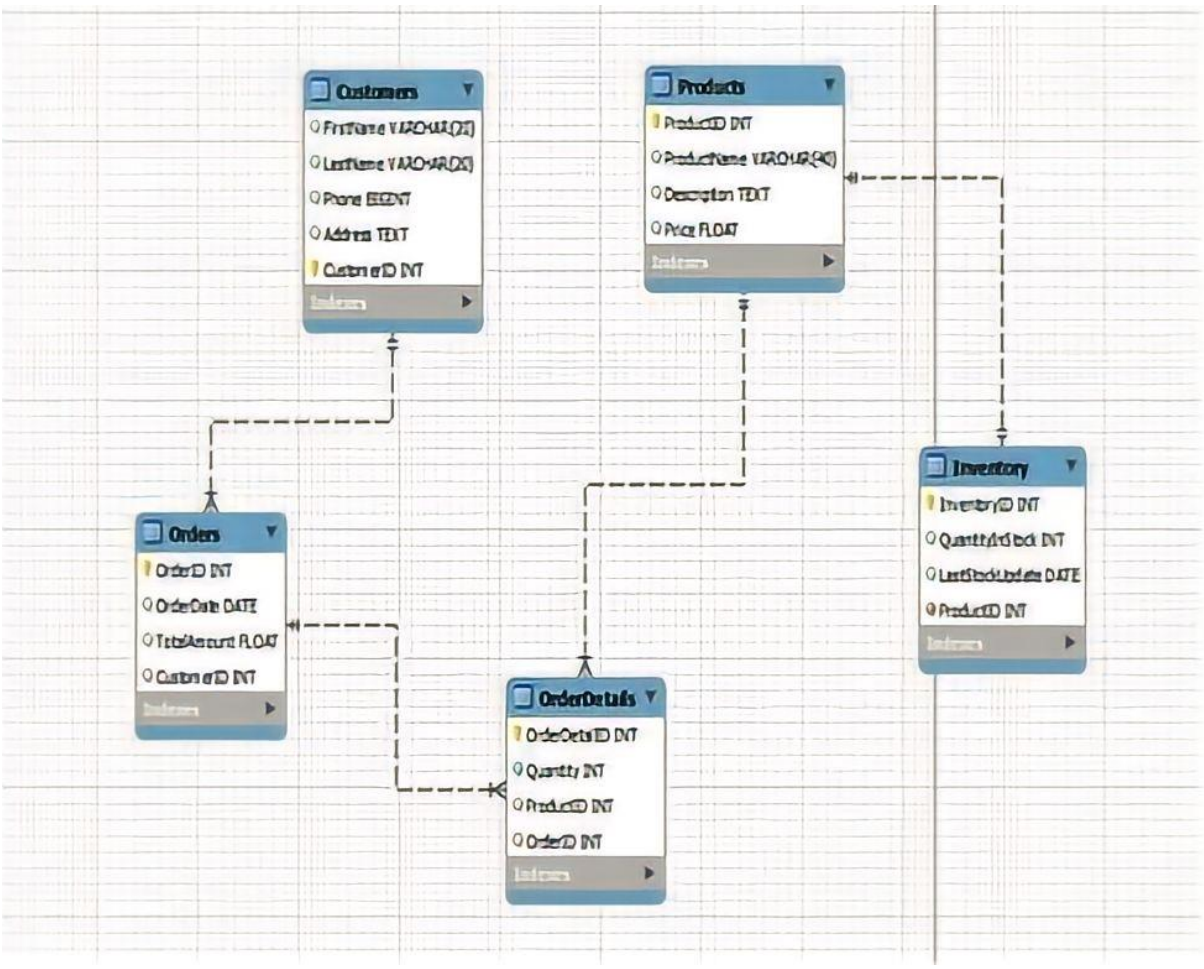
```
mysql> CREATE TABLE Products (
    -> ProductID INT PRIMARY KEY,
    -> ProductName TEXT,
    -> Description TEXT,
    -> Price BIGINT
    -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> CREATE TABLE Orders(
    -> OrderID INT PRIMARY KEY,
    -> CustomerID INT,
    -> OrderDate DATE,
    -> TotalAmount BIGINT,
    -> FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID)
    -> );
Query OK, 0 rows affected (0.09 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 (0.08 sec)
```

```
mysql> CREATE TABLE Inventory (
    -> InventoryID INT,
    -> ProductID INT,
    -> QuantityInStock INT,
    -> LastStockUpdate DATE,
    -> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
Query OK, 0 rows affected (0.09 sec)
```

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



5. Insert at least 10 sample records into each of the following tables.
a. Customers
b. Products
c. Orders
d. OrderDetails

```
mysql> SELECT * FROM Customers;
+------------+-----------+----------+-------------------------+--------------+-------------------------------+
| CustomerID | FirstName | LastName | Email                   | Phone        | Address                       |
+------------+-----------+----------+-------------------------+--------------+-------------------------------+
|          1 | Tom       | Smith    | tomsmith@gmail.com      |   7839567284 | 245 Mumbai, Maharastra        |
|          2 | Kevin     | Smith    | kevinsmith@gmail.com    |    789302284 | 145 Nagpur, Maharastra        |
|          3 | Kelly     | Shell    | kellyshell23@gmail.com  |    122344454 | 20, Medinpur, West Bengal     |
|          4 | Carl      | Good     | carlgoodson11@gmail.com |   8976444454 | 2 Haldia, West Bengal         |
|          5 | Madison   | Bid      | madisonbd234@gmail.com  |   9876543290 | 34 Ullasnagar, Maharastra     |
|          6 | Freddy    | Park     | freddypark348@gmail.com |   8940973458 | 87 Newtown, West Bengal       |
|          7 | Joy       | Miller   | joymillerks123@gmail.com|   5809456378 | 980 Newtown, West Bengal      |
|          8 | James     | Pitt     | jamespitt657@gmail.com  |  98056278402 | 90 Haldia, West Bengal        |
|          9 | Jiya      | Flower   | jiyaflower3578@gmail.com| 786542399860 | 67 Chandnichowk, New Delhi    |
|         10 | Susan     | Hill     | susankihill567@gmail.com| 678955667788 | 380 Sarojini Nagar, New Delhi |
+------------+-----------+----------+-------------------------+--------------+-------------------------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM Products;
+-----------+------------------+---------------------------+--------+
| ProductID | ProductName      | Description               | Price  |
+-----------+------------------+---------------------------+--------+
|        10 | Asus Laptop      | 16GB Ryzen7700X 1TB       |  58000 |
|        12 | HP Laptop        | 16GB i5-13Gen 1TB         |  80000 |
|        13 | Lenovo Keyboard  | Mec Bl-S                  |   5000 |
|        14 | Logitexh Mouse   | G402 wired                |   3000 |
|        15 | Lenovo Yoga      | 18GB i7-15Gen 4TB         | 180000 |
|        16 | Razor V3         | BT-6e wired               |   2500 |
|        17 | Xbox s-controller| 500mAh wireless dual-shock|   8000 |
|        18 | Boat Airdopes    | 190 TWS 30hr              |   2000 |
|        19 | Asus ROG Ally    | 16GB AMD-Ryzen-Z1 1TB     |  60000 |
|        20 | HP Omen 16       | 8GB Ryzen-7-RTX-3070 1TB  | 108000 |
+-----------+------------------+---------------------------+--------+
```

```
mysql> SELECT * FROM Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|    1111 |          6 | 2024-02-15 |       58000 |
|    1112 |          7 | 2020-12-25 |       58000 |
|    1113 |          9 | 2013-10-04 |        5000 |
|    1114 |          1 | 2020-11-08 |        2000 |
|    1115 |          7 | 2023-10-25 |       58000 |
|    1116 |          4 | 2020-11-11 |        2000 |
|    1117 |          4 | 2023-12-01 |        8000 |
|    1118 |         10 | 2023-12-20 |      108000 |
|    1119 |          7 | 2023-12-20 |       80000 |
|    1120 |          9 | 2024-01-20 |        5000 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|             1 |    1111 |        10 |        1 |
|             2 |    1112 |        10 |        1 |
|             3 |    1113 |        13 |        2 |
|             4 |    1114 |        18 |        1 |
|             5 |    1115 |        10 |        1 |
|             6 |    1116 |        18 |        1 |
|             7 |    1117 |        17 |        1 |
|             8 |    1118 |        20 |        1 |
|             9 |    1119 |        12 |        1 |
|            10 |    1120 |        13 |        1 |
+---------------+---------+-----------+----------+
10 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM Inventory;
+-------------+-----------+-----------------+-----------------+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-------------+-----------+-----------------+-----------------+
|         211 |        10 |               3 | 2023-10-25      |
|         212 |        12 |               4 | 2023-12-20      |
|         213 |        13 |               3 | 2024-01-20      |
|         214 |        14 |               5 | 2013-01-01      |
|         215 |        15 |               5 | 2013-01-01      |
|         216 |        16 |               5 | 2013-01-01      |
|         217 |        17 |               4 | 2023-12-01      |
|         218 |        18 |               3 | 2020-11-11      |
|         219 |        19 |               5 | 2013-01-01      |
|         220 |        20 |               4 | 2023-12-20      |
+-------------+-----------+-----------------+-----------------+
10 rows in set (0.00 sec)
```

Tasks 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                   |
+-----------+----------+-------------------------+
| Tom       | Smith    | tomsmith@gmail.com      |
| Kevin     | Smith    | kevinsmith@gmail.com    |
| Kelly     | Shell    | kellyshell23@gmail.com  |
| Carl      | Good     | carlgoodson11@gmail.com |
| Madison   | Bid      | madisonbd234@gmail.com  |
| Freddy    | Park     | freddypark348@gmail.com |
| Joy       | Miller   | joymillerks123@gmail.com|
| James     | Pitt     | jamespitt657@gmail.com  |
| Jiya      | Flower   | jiyaflower3578@gmail.com|
| Susan     | Hill     | susankihill567@gmail.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 OrderID, OrderDate, (SELECT CONCAT(FirstName,' ',LastName)
    -> FROM Customers WHERE CustomerID = Orders.CustomerID) AS CustomerName
    -> FROM Orders;
+---------+------------+--------------+
| OrderID | OrderDate  | CustomerName |
+---------+------------+--------------+
|    1111 | 2024-02-15 | Freddy Park  |
|    1112 | 2020-12-25 | Joy Miller   |
|    1113 | 2013-10-04 | Jiya Flower  |
|    1114 | 2020-11-08 | Tom Smith    |
|    1115 | 2023-10-25 | Joy Miller   |
|    1116 | 2020-11-11 | Carl Good    |
|    1117 | 2023-12-01 | Carl Good    |
|    1118 | 2023-12-20 | Susan Hill   |
|    1119 | 2023-12-20 | Joy Miller   |
|    1120 | 2024-01-20 | Jiya Flower  |
+---------+------------+--------------+
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, 'Jim', 'Kelp', 'jimmykelp@gmail.com', 76249986645, '245 Ranchi, Jharkhand');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Customers;
+------------+-----------+----------+-------------------------+-------------+-------------------------------+
| CustomerID | FirstName | LastName | Email                   | Phone       | Address                       |
+------------+-----------+----------+-------------------------+-------------+-------------------------------+
|          1 | Tom       | Smith    | tomsmith@gmail.com      |  7839567284 | 245 Mumbai, Maharastra        |
|          2 | Kevin     | Smith    | kevinsmith@gmail.com    |   789302284 | 145 Nagpur, Maharastra        |
|          3 | Kelly     | Shell    | kellyshell23@gmail.com  |   122344454 | 20, Medinpur, West Bengal     |
|          4 | Carl      | Good     | carlgoodson11@gmail.com |  8976444454 | 2 Haldia, West Bengal         |
|          5 | Madison   | Bid      | madisonbd234@gmail.com  |  9876543290 | 34 Ullasnagar, Maharastra     |
|          6 | Freddy    | Park     | freddypark348@gmail.com |  8940973458 | 87 Newtown, West Bengal       |
|          7 | Joy       | Miller   | joymillerks123@gmail.com |  5809456378 | 980 Newtown, West Bengal      |
|          8 | James     | Pitt     | jamespitt657@gmail.com  | 98056278402 | 90 Haldia, West Bengal        |
|          9 | Jiya      | Flower   | jiyaflower3578@gmail.com | 786542399860 | 67 Chandnichowk, New Delhi    |
|         10 | Susan     | Hill     | susankihill567@gmail.com | 678955667788 | 380 Sarojini Nagar, New Delhi |
|         11 | Jim       | Kelp     | jimmykelp@gmail.com     | 76249986645 | 245 Ranchi, Jharkhand         |
+------------+-----------+----------+-------------------------+-------------+-------------------------------+
11 rows in set (0.00 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  |
+-----------+-------------------+---------------------------+--------+
|        10 | Asus Laptop       | 16GB Ryzen7700X 1TB       |  63800 |
|        12 | HP Laptop         | 16GB i5-13Gen 1TB         |  88000 |
|        13 | Lenovo Keyboard   | Mec Bl-S                  |   5500 |
|        14 | Logitexh Mouse    | G402 wired                |   3300 |
|        15 | Lenovo Yoga       | 18GB i7-15Gen 4TB         | 198000 |
|        16 | Razor V3          | BT-6e wired               |   2750 |
|        17 | Xbox s-controller | 500mAh wireless dual-shock |   8800 |
|        18 | Boat Airdopes     | 190 TWS 30hr              |   2200 |
|        19 | Asus ROG Ally     | 16GB AMD-Ryzen-Z1 1TB     |  66000 |
|        20 | HP Omen 16        | 8GB Ryzen-7-RTX-3070 1TB  | 118800 |
+-----------+-------------------+---------------------------+--------+
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> SET @OrderID = 1111;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM OrderDetails
    -> WHERE OrderID = @OrderID;
Query OK, 1 row affected (0.02 sec)

mysql> DELETE FROM Orders
    -> WHERE OrderID = @OrderID;
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> INSERT INTO Orders(OrderID, CustomerID, OrderDate, TotalAmount)
    -> VALUES (1121, 6, '2022-12-09', 80000);
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

```
mysql> SET @CustomerID = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SET @Email = 'tommysmith23@gmail.com';
Query OK, 0 rows affected (0.00 sec)

mysql> SET @Address = '267 Park Street, West Bengal';
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE Customers
    -> SET Email = @Email, Address = @address
    -> WHERE CustomerID = @CustomerID;
Query OK, 1 row affected (0.01 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 * TotalAmount)
    -> FROM OrderDetails
    -> WHERE OrderDetails.OrderID = Orders.OrderID);
Query OK, 2 rows affected (0.01 sec)
Rows matched: 10  Changed: 2  Warnings: 0

mysql> SELECT * FROM Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|    1112 |          7 | 2020-12-25 |       58000 |
|    1113 |          9 | 2013-10-04 |       10000 |
|    1114 |          1 | 2020-11-08 |        2000 |
|    1115 |          7 | 2023-10-25 |       58000 |
|    1116 |          4 | 2020-11-11 |        2000 |
|    1117 |          4 | 2023-12-01 |        8000 |
|    1118 |         10 | 2023-12-20 |      108000 |
|    1119 |          7 | 2023-12-20 |       80000 |
|    1120 |          9 | 2024-01-20 |        5000 |
|    1121 |          6 | 2022-12-09 |        NULL |
+---------+------------+------------+-------------+
10 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> DELETE FROM OrderDetails
    -> WHERE OrderID IN (
    -> SELECT OrderID
    -> FROM Orders
    -> WHERE CustomerID = @CustomerID
    -> );
Query OK, 2 rows affected (0.01 sec)

mysql> DELETE FROM Orders
    -> WHERE CustomerID = @CustomerID;
Query OK, 2 rows affected (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> INSERT INTO Products
    -> VALUES (11, 'Dell ALienware', '16GB i7-13Gen 1TB', 75000);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Products;
+-----------+------------------+--------------------------+--------+
| ProductID | ProductName      | Description              | Price  |
+-----------+------------------+--------------------------+--------+
|        10 | Asus Laptop      | 16GB Ryzen7700X 1TB      |  63800 |
|        11 | Dell ALienware   | 16GB i7-13Gen 1TB        |  75000 |
|        12 | HP Laptop        | 16GB i5-13Gen 1TB        |  88000 |
|        13 | Lenovo Keyboard  | Mec Bl-S                 |   5500 |
|        14 | Logitexh Mouse   | G402 wired               |   3300 |
|        15 | Lenovo Yoga      | 18GB i7-15Gen 4TB        | 198000 |
|        16 | Razor V3         | BT-6e wired              |   2750 |
|        17 | Xbox s-controller| 500mAh wireless dual-shock |   8800 |
|        18 | Boat Airdopes    | 190 TWS 30hr             |   2200 |
|        19 | Asus ROG Ally    | 16GB AMD-Ryzen-Z1 1TB    |  66000 |
|        20 | HP Omen 16       | 8GB Ryzen-7-RTX-3070 1TB | 118800 |
+-----------+------------------+--------------------------+--------+
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> SET @OrderID = 1118;
Query OK, 0 rows affected (0.00 sec)

mysql> SET @NewStatus = 'Shipped';
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE Orders
    -> SET Status = @NewStatus
    -> WHERE OrderID = @OrderID;
Query OK, 1 row affected (0.01 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 Orders.CustomerID = Customers.CustomerID
    -> );
Query OK, 11 rows affected (0.01 sec)
Rows matched: 11  Changed: 11  Warnings: 0
```

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

```
mysql> SELECT Orders.OrderID, Orders.OrderDate,
    -> Customers.FirstName, Customers.LastName
    -> FROM Orders
    -> JOIN Customers
    -> ON Orders.CustomerID = Customers.CustomerID;
+---------+------------+-----------+----------+
| OrderID | OrderDate  | FirstName | LastName |
+---------+------------+-----------+----------+
|    1112 | 2020-12-25 | Joy       | Miller   |
|    1114 | 2020-11-08 | Tom       | Smith    |
|    1115 | 2023-10-25 | Joy       | Miller   |
|    1116 | 2020-11-11 | Carl      | Good     |
|    1117 | 2023-12-01 | Carl      | Good     |
|    1118 | 2023-12-20 | Susan     | Hill     |
|    1119 | 2023-12-20 | Joy       | Miller   |
|    1121 | 2022-12-09 | Freddy    | Park     |
+---------+------------+-----------+----------+
8 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,
    -> COALESCE(SUM(OD.Quantity * P.Price), 0) AS TotalRevenue
    -> FROM Products P
    -> LEFT JOIN OrderDetails OD
    -> ON P.ProductID = OD.ProductID
    -> LEFT JOIN Orders O
    -> ON OD.OrderID = O.OrderID
    -> GROUP BY P.ProductID, P.ProductName;
+-----------+------------------+--------------+
| ProductID | ProductName      | TotalRevenue |
+-----------+------------------+--------------+
|        10 | Asus Laptop      |       127600 |
|        11 | Dell ALienware   |            0 |
|        12 | HP Laptop        |        88000 |
|        13 | Lenovo Keyboard  |            0 |
|        14 | Logitexh Mouse   |            0 |
|        15 | Lenovo Yoga      |            0 |
|        16 | Razor V3         |            0 |
|        17 | Xbox s-controller |         8800 |
|        18 | Boat Airdopes    |         4400 |
|        19 | Asus ROG Ally    |            0 |
|        20 | HP Omen 16       |       118800 |
+-----------+------------------+--------------+
11 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.FirstName,
    -> Customers.LastName, Customers.Email, Customers.Phone
    -> FROM Customers
    -> JOIN Orders
    -> ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.Lastname;
+-----------+----------+-------------------------+--------------+
| FirstName | LastName | Email                   | Phone        |
+-----------+----------+-------------------------+--------------+
| Tom       | Smith    | tommysmith23@gmail.com  |   7839567284 |
| Carl      | Good     | carlgoodson11@gmail.com |   8976444454 |
| Freddy    | Park     | freddypark348@gmail.com |   8940973458 |
| Joy       | Miller   | joymillerks123@gmail.com |   5809456378 |
| Susan     | Hill     | susankihill567@gmail.com | 678955667788 |
+-----------+----------+-------------------------+--------------+
5 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 P.ProductName AS PopularGadget,
    -> P.OrderQuantity AS Total_Quantity_Ordered
    -> FROM Products P
    -> ORDER BY Total_Quantity_Ordered DESC
    -> LIMIT 1;
+---------------+-------------------------+
| PopularGadget | Total_Quantity_Ordered |
+---------------+-------------------------+
| Asus Laptop   |                       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, Category
    -> FROM Products
    -> ORDER BY ProductID, Category;
+-----------+------------------+------------+
| ProductID | ProductName      | Category   |
+-----------+------------------+------------+
|        10 | Asus Laptop      | Laptop     |
|        11 | Dell ALienware   | Laptop     |
|        12 | HP Laptop        | Laptop     |
|        13 | Lenovo Keyboard  | Keyboard   |
|        14 | Logitexh Mouse   | Mouse      |
|        15 | Lenovo Yoga      | Laptop     |
|        16 | Razor V3         | Headphones |
|        17 | Xbox s-controller | Gamepad    |
|        18 | Boat Airdopes    | Headphones |
|        19 | Asus ROG Ally    | Laptop     |
|        20 | HP Omen 16       | Laptop     |
+-----------+------------------+------------+
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,
    ->     IFNULL(AVG(O.TotalAmount), 0) AS AverageOrderValue
    -> FROM
    ->     Customers C
    -> LEFT JOIN
    ->     Orders O ON C.CustomerID = O.CustomerID
    -> GROUP BY
    ->     C.CustomerID, C.FirstName, C.LastName
    -> ORDER BY
    ->     C.CustomerID;
+------------+-----------+----------+-------------------+
| CustomerID | FirstName | LastName | AverageOrderValue |
+------------+-----------+----------+-------------------+
|          1 | Tom       | Smith    |         2000.0000 |
|          2 | Kevin     | Smith    |            0.0000 |
|          3 | Kelly     | Shell    |            0.0000 |
|          4 | Carl      | Good     |         5000.0000 |
|          5 | Madison   | Bid      |            0.0000 |
|          6 | Freddy    | Park     |        58000.0000 |
|          7 | Joy       | Miller   |        65333.3333 |
|          8 | James     | Pitt     |            0.0000 |
|          9 | Jiya      | Flower   |            0.0000 |
|         10 | Susan     | Hill     |       108000.0000 |
|         11 | Jim       | Kelp     |            0.0000 |
+------------+-----------+----------+-------------------+
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> SELECT
    ->     O.OrderID,
    ->     O.CustomerID,
    ->     C.FirstName,
    ->     C.LastName,
    ->     C.Email,
    ->     C.Phone,
    ->     C.Address,
    ->     SUM(OD.Quantity * P.Price) AS TotalRevenue
    -> FROM
    ->     Orders O
    -> JOIN
    ->     Customers C ON O.CustomerID = C.CustomerID
    -> JOIN
    ->     OrderDetails OD ON O.OrderID = OD.OrderID
    -> JOIN
    ->     Products P ON OD.ProductID = P.ProductID
    -> GROUP BY
    ->     O.OrderID, O.CustomerID, C.FirstName, C.LastName, C.Email, C.Phone, C.Address
    -> ORDER BY
    ->     TotalRevenue DESC
    -> LIMIT 1;
```

```
+---------+------------+-----------+----------+------------------------+-------------+-----------------------------+--------------+
| OrderID | CustomerID | FirstName | LastName | Email                  | Phone       | Address                     | TotalRevenue |
+---------+------------+-----------+----------+------------------------+-------------+-----------------------------+--------------+
|    1118 |         10 | Susan     | Hill     | susankihill567@gmail.com | 678955667788 | 380 Sarojini Nagar, New Delhi |       118800 |
+---------+------------+-----------+----------+------------------------+-------------+-----------------------------+--------------+
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,
    ->     P.Category,
    ->     COUNT(OD.OrderID) AS OrderCount
    -> FROM
    ->     Products P
    -> LEFT JOIN
    ->     OrderDetails OD ON P.ProductID = OD.ProductID
    -> GROUP BY
    ->     P.ProductID, P.ProductName, P.Category;
+-----------+-------------------+------------+------------+
| ProductID | ProductName       | Category   | OrderCount |
+-----------+-------------------+------------+------------+
|        10 | Asus Laptop       | Laptop     |          3 |
|        11 | Dell ALienware    | Laptop     |          0 |
|        12 | HP Laptop         | Laptop     |          1 |
|        13 | Lenovo Keyboard   | Keyboard   |          0 |
|        14 | Logitexh Mouse    | Mouse      |          0 |
|        15 | Lenovo Yoga       | Laptop     |          0 |
|        16 | Razor V3          | Headphones |          0 |
|        17 | Xbox s-controller | Gamepad    |          1 |
|        18 | Boat Airdopes     | Headphones |          2 |
|        19 | Asus ROG Ally     | Laptop     |          0 |
|        20 | HP Omen 16        | Laptop     |          1 |
+-----------+-------------------+------------+------------+
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> DELIMITER //
mysql> CREATE PROCEDURE FindCustomer(IN input_product_name VARCHAR(225))
    -> BEGIN
    -> SELECT C.CustomerID, C.FirstName, C.LastName,
    -> C.Email, C.Phone, C.Address
    -> FROM Customers C
    -> JOIN Orders O ON O.CustomerID = C.CustomerID
    -> JOIN OrderDetails OD ON OD.OrderID = O.OrderID
    -> JOIN Products P ON P.ProductID = OD.ProductID
    -> WHERE P.ProductName = input_product_name;
    -> END //
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER ;
mysql> CALL FindCustomer('HP Laptop');
+------------+-----------+----------+-------------------------+------------+--------------------------+
| CustomerID | FirstName | LastName | Email                   | Phone      | Address                  |
+------------+-----------+----------+-------------------------+------------+--------------------------+
|          7 | Joy       | Miller   | joymillerks123@gmail.com | 5809456378 | 980 Newtown, West Bengal |
+------------+-----------+----------+-------------------------+------------+--------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 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 P.ProductID, P.ProductName,
    -> SUM(OD.Quantity * P.Price) AS TotalProductRevenue
    -> FROM Orders O
    -> JOIN OrderDetails OD ON O.OrderID = OD.OrderID
    -> JOIN Products P ON P.ProductID = OD.ProductID
    -> GROUP BY P.ProductID, P.ProductName;
+-----------+-------------------+----------------------+
| ProductID | ProductName       | TotalProductRevenue |
+-----------+-------------------+----------------------+
|        10 | Asus Laptop       |               191400 |
|        18 | Boat Airdopes     |                 4400 |
|        17 | Xbox s-controller |                 8800 |
|        20 | HP Omen 16        |               118800 |
|        12 | HP Laptop         |                88000 |
+-----------+-------------------+----------------------+
5 rows in set (0.00 sec)
```

Task 4. Subquery and its type:

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

```
mysql> SELECT CustomerID, FirstName, LastName, Email, Phone,
    -> Address
    -> FROM Customers
    -> WHERE CustomerID NOT IN (SELECT CustomerID FROM Orders WHERE CustomerID IS NOT NULL);
+------------+-----------+----------+-------------------------+--------------+---------------------------+
| CustomerID | FirstName | LastName | Email                   | Phone        | Address                   |
+------------+-----------+----------+-------------------------+--------------+---------------------------+
|          2 | Kevin     | Smith    | kevinsmith@gmail.com    |    789302284 | 145 Nagpur, Maharastra    |
|          3 | Kelly     | Shell    | kellyshell23@gmail.com  |    122344454 | 20, Medinpur, West Bengal |
|          5 | Madison   | Bid      | madisonbd234@gmail.com  |   9876543290 | 34 Ullasnagar, Maharastra |
|          8 | James     | Pitt     | jamespitt657@gmail.com  |  98056278402 | 90 Haldia, West Bengal    |
|          9 | Jiya      | Flower   | jiyaflower3578@gmail.com | 786542399860 | 67 Chandnichowk, New Delhi |
|         11 | Jim       | Kelp     | jimmykelp@gmail.com     |  76249986645 | 245 Ranchi, Jharkhand     |
+------------+-----------+----------+-------------------------+--------------+---------------------------+
6 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 (0.02 sec)
```

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

```
mysql> SELECT SUM(OD.Quantity * P.Price) AS TotalRevenue
    -> FROM OrderDetails OD
    -> JOIN Products P ON OD.ProductID = P.ProductID;
+--------------+
| TotalRevenue |
+--------------+
|       411400 |
+--------------+
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> DELIMITER //
mysql> CREATE PROCEDURE CalculateAverageQuantity(IN input_category VARCHAR(225))
    -> BEGIN
    -> SELECT
    -> AVG(OD.Quantity) AS AverageQuantity
    -> FROM OrderDetails OD
    -> JOIN Products P ON OD.ProductID = P.ProductID
    -> WHERE P.Category = input_category;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> CALL CalculateAverageQuantity('Laptop');
+-----------------+
| AverageQuantity |
+-----------------+
|          1.0000 |
+-----------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (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> DELIMITER //
mysql> CREATE PROCEDURE CalculateTotalRevenue(IN input_customer_id INT)
    -> BEGIN
    -> SELECT SUM(OD.Quantity * P.Price) AS TotalRevenue
    -> FROM OrderDetails OD
    -> JOIN Products P ON OD.ProductID = P.ProductID
    -> JOIN Orders O ON OD.OrderID = O.OrderID
    -> WHERE O.CustomerID = input_customer_id;
    -> END //
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER ;
mysql> CALL CalculateTotalRevenue(7);
+--------------+
| TotalRevenue |
+--------------+
|       215600 |
+--------------+
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 C
    -> JOIN Orders O ON C.CustomerID = O.CustomerID
    -> GROUP BY C.CustomerID, C.FirstName, C.LastName
    -> ORDER BY NumberOfOrders DESC;
+------------+-----------+----------+----------------+
| CustomerID | FirstName | LastName | NumberOfOrders |
+------------+-----------+----------+----------------+
|          7 | Joy       | Miller   |              3 |
|          4 | Carl      | Good     |              2 |
|          1 | Tom       | Smith    |              1 |
|          6 | Freddy    | Park     |              1 |
|         10 | Susan     | Hill     |              1 |
+------------+-----------+----------+----------------+
5 rows 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.Category,
    -> SUM(OD.Quantity) AS TotalQuantityOrdered
    -> FROM OrderDetails OD
    -> JOIN Products P ON OD.ProductID = P.ProductID
    -> GROUP BY P.Category
    -> ORDER BY TotalQuantityOrdered DESC
    -> LIMIT 1;
+----------+----------------------+
| Category | TotalQuantityOrdered |
+----------+----------------------+
| Laptop   |                    5 |
+----------+----------------------+
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(OD.Quantity * P.Price) AS TotalSpending
    -> FROM Customers C
    -> JOIN Orders O ON C.CustomerID = O.CustomerID
    -> JOIN OrderDetails OD ON O.OrderID = OD.OrderID
    -> JOIN Products P ON OD.ProductID = P.ProductID
    -> GROUP BY C.CustomerID, C.FirstName, C.LastName
    -> ORDER BY TotalSpending DESC
    -> LIMIT 1;
+------------+-----------+----------+---------------+
| CustomerID | FirstName | LastName | TotalSpending |
+------------+-----------+----------+---------------+
|          7 | Joy       | Miller   |        215600 |
+------------+-----------+----------+---------------+
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(TotalRevenue / NumberOfOrders) AS AverageOrderValue
    -> FROM (
    -> SELECT O.CustomerID,
    -> COUNT(DISTINCT O.OrderID) AS NumberOfOrders,
    -> SUM(OD.Quantity * P.Price) AS TotalRevenue
    -> FROM Orders O
    -> JOIN OrderDetails OD ON O.OrderID = OD.OrderID
    -> JOIN Products P ON OD.ProductID = P.ProductID
    -> GROUP BY O.CustomerID
    -> ) AS CustomerOrderStatus;
+-------------------+
| AverageOrderValue |
+-------------------+
|    52433.33333333 |
+-------------------+
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 C
    -> LEFT JOIN
    -> Orders O ON C.CustomerID = O.CustomerID
    -> GROUP BY
    -> C.CustomerID, C.FirstName, C.LastName;
+------------+-----------+----------+------------+
| CustomerID | FirstName | LastName | OrderCount |
+------------+-----------+----------+------------+
|          1 | Tom       | Smith    |          1 |
|          2 | Kevin     | Smith    |          0 |
|          3 | Kelly     | Shell    |          0 |
|          4 | Carl      | Good     |          2 |
|          5 | Madison   | Bid      |          0 |
|          6 | Freddy    | Park     |          1 |
|          7 | Joy       | Miller   |          3 |
|          8 | James     | Pitt     |          0 |
|          9 | Jiya      | Flower   |          0 |
|         10 | Susan     | Hill     |          1 |
|         11 | Jim       | Kelp     |          0 |
+------------+-----------+----------+------------+
11 rows in set (0.00 sec)
```