TECHSHOP

Task:1. Database Design:

1. Create the database named "TechShop"

```
CREATE DATABASE TECHSHOP;

USE DATABASE TECHSHOP;
```

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

1. Customers:

- CustomerID (Primary Key)
- FirstName
- LastName
- Email
- Phone
- Address

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Phone VARCHAR(20),
    Address VARCHAR(255));
```

2. Products:

- ProductID (Primary Key)
- ProductName
- Description
- Price

```
CREATE TABLE Products (
    ProductID INT  auto_increment,
    ProductName VARCHAR(100),
    Description TEXT,
    Price int);
```

3. Orders:

- OrderID (Primary Key)
- CustomerID (Foreign Key referencing Customers)
- OrderDate
- TotalAmount

```
CREATE TABLE Orders (
    OrderID INT ,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount INT);
```

4. OrderDetails:

- OrderDetailID (Primary Key)
- OrderID (Foreign Key referencing Orders)
- ProductID (Foreign Key referencing Products)
- Quantity

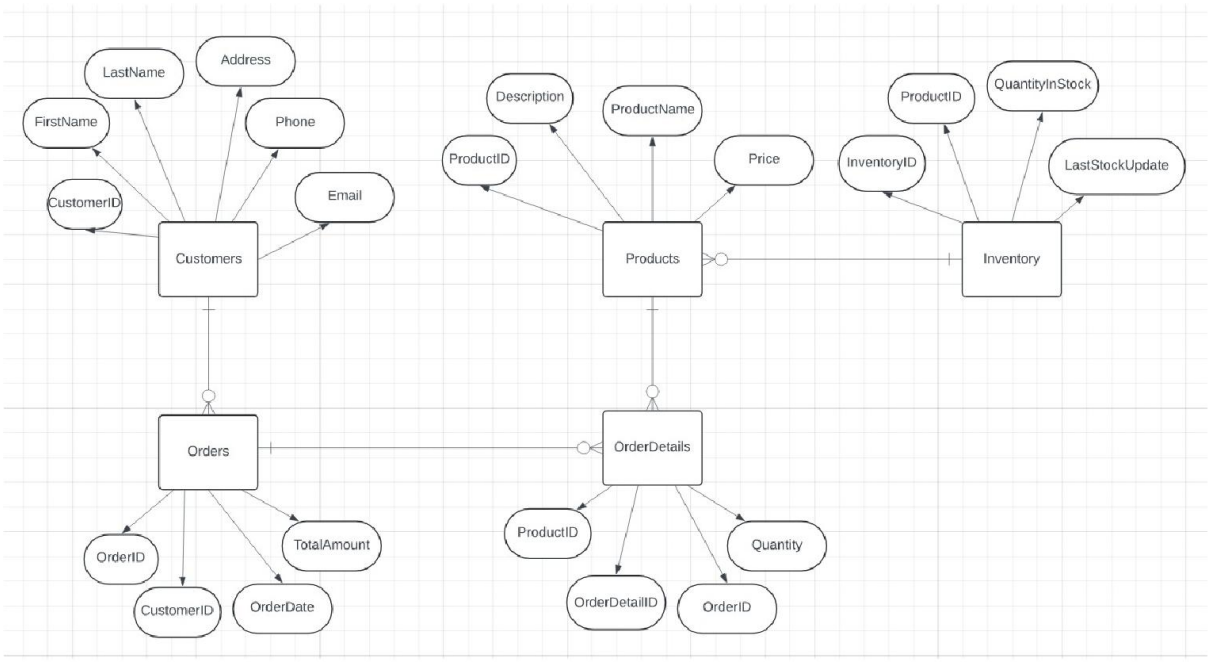
```
CREATE TABLE OrderDetails (
    OrderDetailID INT auto_increment,
    OrderID INT,
    ProductID INT,
    Quantity INT);
```

5. Inventory:

- InventoryID (Primary Key)
- ProductID (Foreign Key referencing Products)
- QuantityInStock
- LastStockUpdate

```
CREATE TABLE Inventory (
    InventoryID INT auto_increment,
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate DATETIME);
```

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

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

VALUES

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

('Jane', 'Smith', 'jane@example.com', '456-789-0123', '456 Elm St'),

('Alice', 'Johnson', 'alice@example.com', '789-012-3456', '789 Oak St'),

('Bob', 'Williams', 'bob@example.com', '234-567-8901', '234 Maple St'),

('Emily', 'Brown', 'emily@example.com', '567-890-1234', '567 Pine St'),

('Michael', 'Jones', 'michael@example.com', '890-123-4567', '890 Cedar St'),

('Sarah', 'Garcia', 'sarah@example.com', '345-678-9012', '345 Birch St'),

('David', 'Martinez', 'david@example.com', '678-901-2345', '678 Walnut St'),

('Jennifer', 'Rodriguez', 'jennifer@example.com', '901-234-5678', '901 Oak St'),

('William', 'Hernandez', 'william@example.com', '123-456-7890', '123 Elm St');

```
mysql> select * from customers;
+------------+-----------+-----------+---------------------+--------------+---------------+
| CustomerID | FirstName | LastName  | Email               | Phone        | Address       |
+------------+-----------+-----------+---------------------+--------------+---------------+
|          1 | John      | Doe       | john@example.com    | 123-456-7890 | 123 Main St   |
|          2 | Jane      | Smith     | jane@example.com    | 456-789-0123 | 456 Elm St    |
|          3 | Alice     | Johnson   | alice@example.com   | 789-012-3456 | 789 Oak St    |
|          4 | Bob       | Williams  | bob@example.com     | 234-567-8901 | 234 Maple St  |
|          5 | Emily     | Brown     | emily@example.com   | 567-890-1234 | 567 Pine St   |
|          6 | Michael   | Jones     | michael@example.com | 890-123-4567 | 890 Cedar St  |
|          7 | Sarah     | Garcia    | sarah@example.com   | 345-678-9012 | 345 Birch St  |
|          8 | David     | Martinez  | david@example.com   | 678-901-2345 | 678 Walnut St |
|          9 | Jennifer  | Rodriguez | jennifer@example.com | 901-234-5678 | 901 Oak St    |
|         10 | William   | Hernandez | william@example.com | 123-456-7890 | 123 Elm St    |
+------------+-----------+-----------+---------------------+--------------+---------------+
10 rows in set (0.00 sec)
```

b. Products

INSERT INTO Products (ProductName, Description, Price)

VALUES

('Laptop', 'High-performance laptop with SSD', 999),

('Smartphone', 'Latest model with dual camera', 699),

('Tablet', '10-inch tablet with touchscreen', 299),

('Smartwatch', 'Fitness tracker with heart rate monitor', 199),

('Headphones', 'Noise-canceling wireless headphones', 149),

('Camera', 'DSLR camera with 18-55mm lens', 799),

('TV', '4K Ultra HD smart TV', 1299),

('Speaker', 'Bluetooth portable speaker', 79),

('Gaming Console', 'Next-gen gaming console', 499),

('Router', 'High-speed Wi-Fi router', 129);

```
mysql> select * from products;
+-----------+---------------+-----------------------------------------+-------+
| ProductID | ProductName   | Description                             | Price |
+-----------+---------------+-----------------------------------------+-------+
|         1 | Laptop        | High-performance laptop with SSD        |   999 |
|         2 | Smartphone    | Latest model with dual camera           |   699 |
|         3 | Tablet        | 10-inch tablet with touchscreen         |   299 |
|         4 | Smartwatch    | Fitness tracker with heart rate monitor |   199 |
|         5 | Headphones    | Noise-canceling wireless headphones     |   149 |
|         6 | Camera        | DSLR camera with 18-55mm lens           |   799 |
|         7 | TV            | 4K Ultra HD smart TV                    |  1299 |
|         8 | Speaker       | Bluetooth portable speaker              |    79 |
|         9 | Gaming Console | Next-gen gaming console                |   499 |
|        10 | Router        | High-speed Wi-Fi router                 |   129 |
+-----------+---------------+-----------------------------------------+-------+
```

c. Orders

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

VALUES

(101,1, '2024-04-01', 999),

(102,2, '2024-04-02', 699),

(103,3, '2024-04-03', 299),

(104,4, '2024-04-04', 199),

(105,5, '2024-04-05', 149),

(106,6, '2024-04-06', 799),

(107,7, '2024-04-07', 1299),

(108,8, '2024-04-08', 79),

(109,9, '2024-04-09', 499),

(110,10, '2024-04-10', 129);

```
mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     101 |          1 | 2024-04-01 |         999 |
|     102 |          2 | 2024-04-02 |         699 |
|     103 |          3 | 2024-04-03 |         299 |
|     104 |          4 | 2024-04-04 |         199 |
|     105 |          5 | 2024-04-05 |         149 |
|     106 |          6 | 2024-04-06 |         799 |
|     107 |          7 | 2024-04-07 |        1299 |
|     108 |          8 | 2024-04-08 |          79 |
|     109 |          9 | 2024-04-09 |         499 |
|     110 |         10 | 2024-04-10 |         129 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

d. OrderDetails

INSERT INTO OrderDetails (OrderID, ProductID, Quantity)

VALUES

(101, 1, 1),

(102, 2, 3),

(103, 3, 2),

(104, 4, 5),

(105, 5, 2),

(106, 6, 1),

(107, 7, 7),

(108, 8, 3),

(109, 9, 6),

(110, 10, 2);

```
mysql> select * from orderdetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|             1 |     101 |         1 |        1 |
|             2 |     102 |         2 |        3 |
|             3 |     103 |         3 |        2 |
|             4 |     104 |         4 |        5 |
|             5 |     105 |         5 |        2 |
|             6 |     106 |         6 |        1 |
|             7 |     107 |         7 |        7 |
|             8 |     108 |         8 |        3 |
|             9 |     109 |         9 |        6 |
|            10 |     110 |        10 |        2 |
+---------------+---------+-----------+----------+
```

e. Inventory

INSERT INTO Inventory (ProductID, QuantityInStock, LastStockUpdate)

VALUES

(1, 10, NOW()),

(2, 20, NOW()),

(3, 15, NOW()),

(4, 30, NOW()),

(5, 25, NOW()),

(6, 5, NOW()),

(7, 8, NOW()),

(8, 12, NOW()),

(9, 3, NOW()),

(10, 18, NOW());

```
mysql> select * from inventory;
+-------------+-----------+-----------------+---------------------+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate     |
+-------------+-----------+-----------------+---------------------+
|           1 |         1 |              10 | 2024-04-13 11:30:54 |
|           2 |         2 |              20 | 2024-04-13 11:30:54 |
|           3 |         3 |              15 | 2024-04-13 11:30:54 |
|           4 |         4 |              30 | 2024-04-13 11:30:54 |
|           5 |         5 |              25 | 2024-04-13 11:30:54 |
|           6 |         6 |               5 | 2024-04-13 11:30:54 |
|           7 |         7 |               8 | 2024-04-13 11:30:54 |
|           8 |         8 |              12 | 2024-04-13 11:30:54 |
|           9 |         9 |               3 | 2024-04-13 11:30:54 |
|          10 |        10 |              18 | 2024-04-13 11:30:54 |
+-------------+-----------+-----------------+---------------------+
```

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

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

```
mysql> select concat(firstname,' ',lastname) as CustomerName,email from customers;
+------------------+--------------------+
| CustomerName     | email              |
+------------------+--------------------+
| John Doe         | john@example.com   |
| Jane Smith       | jane@example.com   |
| Alice Johnson    | alice@example.com  |
| Bob Williams     | bob@example.com    |
| Emily Brown      | emily@example.com  |
| Michael Jones    | michael@example.com|
| Sarah Garcia     | sarah@example.com  |
| David Martinez   | david@example.com  |
| Jennifer Rodriguez | jennifer@example.com |
| William Hernandez | william@example.com |
+------------------+--------------------+
```

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

```
mysql> select orders.orderid,orders.orderdate,
    -> (select concat(firstname,' ',lastname)from customers
    -> where customerid=orders.customerid)as CustomerName
    -> from orders;
+---------+-------------+--------------------+
| orderid | orderdate   | CustomerName       |
+---------+-------------+--------------------+
|     101 | 2024-04-01  | John Doe           |
|     102 | 2024-04-02  | Jane Smith         |
|     103 | 2024-04-03  | Alice Johnson      |
|     104 | 2024-04-04  | Bob Williams       |
|     105 | 2024-04-05  | Emily Brown        |
|     106 | 2024-04-06  | Michael Jones      |
|     107 | 2024-04-07  | Sarah Garcia       |
|     108 | 2024-04-08  | David Martinez     |
|     109 | 2024-04-09  | Jennifer Rodriguez |
|     110 | 2024-04-10  | William Hernandez  |
+---------+-------------+--------------------+
```

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 (FirstName, LastName, Email, Phone, Address)
    -> VALUES ('Emily', 'Taylor', 'emily.t@example.com', '987-654-3210', '789 Maple St');
Query OK, 1 row affected (0.01 sec)

mysql> select * from customers;
+------------+-----------+----------+---------------------+--------------+--------------+
| CustomerID | FirstName | LastName | Email               | Phone        | Address      |
+------------+-----------+----------+---------------------+--------------+--------------+
|          1 | John      | Doe      | john@example.com    | 123-456-7890 | 123 Main St  |
|          2 | Jane      | Smith    | jane@example.com    | 456-789-0123 | 456 Elm St   |
|          3 | Alice     | Johnson  | alice@example.com   | 789-012-3456 | 789 Oak St   |
|          4 | Bob       | Williams | bob@example.com     | 234-567-8901 | 234 Maple St |
|          5 | Emily     | Brown    | emily@example.com   | 567-890-1234 | 567 Pine St  |
|          6 | Michael   | Jones    | michael@example.com | 890-123-4567 | 890 Cedar St |
|          7 | Sarah     | Garcia   | sarah@example.com   | 345-678-9012 | 345 Birch St |
|          8 | David     | Martinez | david@example.com   | 678-901-2345 | 678 Walnut St |
|          9 | Jennifer  | Rodriguez | jennifer@example.com | 901-234-5678 | 901 Oak St  |
|         10 | William   | Hernandez | william@example.com | 123-456-7890 | 123 Elm St  |
|         11 | Emily     | Taylor   | emily.t@example.com | 987-654-3210 | 789 Maple St |
+------------+-----------+----------+---------------------+--------------+--------------+
```

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 (0.01 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from products;
+-----------+---------------+-----------------------------------------+-------+
| ProductID | ProductName   | Description                             | Price |
+-----------+---------------+-----------------------------------------+-------+
|         1 | Laptop        | High-performance laptop with SSD        |  1099 |
|         2 | Smartphone    | Latest model with dual camera           |   769 |
|         3 | Tablet        | 10-inch tablet with touchscreen         |   329 |
|         4 | Smartwatch    | Fitness tracker with heart rate monitor |   219 |
|         5 | Headphones    | Noise-canceling wireless headphones     |   164 |
|         6 | Camera        | DSLR camera with 18-55mm lens           |   879 |
|         7 | TV            | 4K Ultra HD smart TV                    |  1429 |
|         8 | Speaker       | Bluetooth portable speaker              |    87 |
|         9 | Gaming Console | Next-gen gaming console                |   549 |
|        10 | Router        | High-speed Wi-Fi router                 |   142 |
+-----------+---------------+-----------------------------------------+-------+
```

5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables.

```
mysql> DELETE FROM OrderDetails WHERE OrderID = 103;
Query OK, 1 row affected (0.02 sec)

mysql> DELETE FROM Orders WHERE OrderID=103;
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 (111,11, '2024-04-11', 399);
Query OK, 1 row affected (0.01 sec)

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     101 |          1 | 2024-04-01 |         999 |
|     102 |          2 | 2024-04-02 |         699 |
|     103 |          3 | 2024-04-03 |         299 |
|     104 |          4 | 2024-04-04 |         199 |
|     105 |          5 | 2024-04-05 |         149 |
|     106 |          6 | 2024-04-06 |         799 |
|     107 |          7 | 2024-04-07 |        1299 |
|     108 |          8 | 2024-04-08 |          79 |
|     109 |          9 | 2024-04-09 |         499 |
|     110 |         10 | 2024-04-10 |         129 |
|     111 |         11 | 2024-04-11 |         399 |
+---------+------------+------------+-------------+
```

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 firstname='sai',lastname='kosh',Email = 'kosh@example.com', Address = '730 dar St'
    -> where customerid=3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from customers;
+------------+-----------+-----------+---------------------+--------------+---------------+
| CustomerID | FirstName | LastName  | Email               | Phone        | Address       |
+------------+-----------+-----------+---------------------+--------------+---------------+
|          1 | John      | Doe       | john@example.com    | 123-456-7890 | 123 Main St   |
|          2 | Jane      | Smith     | jane@example.com    | 456-789-0123 | 456 Elm St    |
|          3 | sai       | kosh      | kosh@example.com    | 789-012-3456 | 730 dar St    |
|          4 | Bob       | Williams  | bob@example.com     | 234-567-8901 | 234 Maple St  |
|          5 | Emily     | Brown     | emily@example.com   | 567-890-1234 | 567 Pine St   |
|          6 | Michael   | Jones     | michael@example.com | 890-123-4567 | 890 Cedar St  |
|          7 | Sarah     | Garcia    | sarah@example.com   | 345-678-9012 | 345 Birch St  |
|          8 | David     | Martinez  | david@example.com   | 678-901-2345 | 678 Walnut St |
|          9 | Jennifer  | Rodriguez | jennifer@example.com| 901-234-5678 | 901 Oak St    |
|         10 | William   | Hernandez | william@example.com | 123-456-7890 | 123 Elm St    |
|         11 | Emily     | Taylor    | emily.t@example.com | 987-654-3210 | 789 Maple St  |
+------------+-----------+-----------+---------------------+--------------+---------------+
11 rows in set (0.00 sec)
```

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

```
mysql> UPDATE Orders
    -> SET TotalAmount = (
    ->     SELECT SUM(Quantity * Price)
    ->     FROM OrderDetails
    ->     JOIN Products ON OrderDetails.ProductID = Products.ProductID
    ->     WHERE OrderDetails.OrderID = Orders.OrderID
    -> );
Query OK, 11 rows affected (0.02 sec)
Rows matched: 11  Changed: 11  Warnings: 0

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     101 |          1 | 2024-04-01 |        1099 |
|     102 |          2 | 2024-04-02 |        2307 |
|     103 |          3 | 2024-04-03 |         658 |
|     104 |          4 | 2024-04-04 |        1095 |
|     105 |          5 | 2024-04-05 |         328 |
|     106 |          6 | 2024-04-06 |         879 |
|     107 |          7 | 2024-04-07 |       10003 |
|     108 |          8 | 2024-04-08 |         261 |
|     109 |          9 | 2024-04-09 |        3294 |
|     110 |         10 | 2024-04-10 |         284 |
|     111 |         11 | 2024-04-11 |        NULL |
+---------+------------+------------+-------------+
```

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> Set @customerid=10;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM OrderDetails WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = @customerid);
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM Orders WHERE CustomerID = @customerid;
Query OK, 1 row affected (0.01 sec)

mysql> Select * From orderdetails;
+--------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+--------------+---------+-----------+----------+
|            1 |     101 |         1 |        1 |
|            2 |     102 |         2 |        3 |
|            4 |     104 |         4 |        0 |
|            5 |     105 |         5 |        2 |
|            6 |     106 |         6 |        1 |
|            7 |     107 |         7 |        7 |
|            8 |     108 |         8 |        3 |
|            9 |     109 |         9 |        0 |
+--------------+---------+-----------+----------+
```

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 (ProductName, Description, Price)
    -> VALUES ('Smart Speaker', 'Voice-controlled smart speaker',749);
Query OK, 1 row affected (0.01 sec)

mysql> select * from products;
+-----------+----------------+-----------------------------------------+-------+
| ProductID | ProductName    | Description                             | Price |
+-----------+----------------+-----------------------------------------+-------+
|         1 | Laptop         | High-performance laptop with SSD        |  1099 |
|         2 | Smartphone     | Latest model with dual camera           |   769 |
|         3 | Tablet         | 10-inch tablet with touchscreen         |   329 |
|         4 | Smartwatch     | Fitness tracker with heart rate monitor |   219 |
|         5 | Headphones     | Noise-canceling wireless headphones     |   164 |
|         6 | Camera         | DSLR camera with 18-55mm lens           |   879 |
|         7 | TV             | 4K Ultra HD smart TV                    |  1429 |
|         8 | Speaker        | Bluetooth portable speaker              |    87 |
|         9 | Gaming Console | Next-gen gaming console                 |   549 |
|        10 | Router         | High-speed Wi-Fi router                 |   142 |
|        11 | Smart Speaker  | Voice-controlled smart speaker          |   749 |
+-----------+----------------+-----------------------------------------+-------+
```

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 status='shipped'
    -> where orderid in (103,105,101,109);
Query OK, 4 rows affected (0.01 sec)
Rows matched: 4  Changed: 4  Warnings: 0

mysql> select * from orders;
+---------+------------+------------+-------------+---------+
| OrderID | CustomerID | OrderDate  | TotalAmount | Status  |
+---------+------------+------------+-------------+---------+
|     101 |          1 | 2024-04-01 |        1099 | shipped |
|     102 |          2 | 2024-04-02 |        2307 | pending |
|     103 |          3 | 2024-04-03 |         658 | shipped |
|     104 |          4 | 2024-04-04 |        1095 | pending |
|     105 |          5 | 2024-04-05 |         328 | shipped |
|     106 |          6 | 2024-04-06 |         879 | pending |
|     107 |          7 | 2024-04-07 |       10003 | pending |
|     108 |          8 | 2024-04-08 |         261 | pending |
|     109 |          9 | 2024-04-09 |        3294 | shipped |
|     110 |         10 | 2024-04-10 |         284 | pending |
|     111 |         11 | 2024-04-11 |        NULL | pending |
+---------+------------+------------+-------------+---------+
```

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 quantity
    -> from orderdetails
    -> where customers.customerid=orderdetails.orderdetailid);
Query OK, 10 rows affected (0.01 sec)
Rows matched: 11  Changed: 10  Warnings: 0

mysql> select * from customers;
+------------+-----------+-----------+---------------------+--------------+---------------+----------------+
| CustomerID | FirstName | LastName  | Email               | Phone        | Address       | NumberofOrders |
+------------+-----------+-----------+---------------------+--------------+---------------+----------------+
|          1 | John      | Doe       | john@example.com    | 123-456-7890 | 123 Main St   |              1 |
|          2 | Jane      | Smith     | jane@example.com    | 456-789-0123 | 456 Elm St    |              3 |
|          3 | sai       | kosh      | kosh@example.com    | 789-012-3456 | 730 dar St    |              2 |
|          4 | Bob       | Williams  | bob@example.com     | 234-567-8901 | 234 Maple St  |              5 |
|          5 | Emily     | Brown     | emily@example.com   | 567-890-1234 | 567 Pine St   |              2 |
|          6 | Michael   | Jones     | michael@example.com | 890-123-4567 | 890 Cedar St  |              1 |
|          7 | Sarah     | Garcia    | sarah@example.com   | 345-678-9012 | 345 Birch St  |              7 |
|          8 | David     | Martinez  | david@example.com   | 678-901-2345 | 678 Walnut St |              3 |
|          9 | Jennifer  | Rodriguez | jennifer@example.com | 901-234-5678 | 901 Oak St    |              6 |
|         10 | William   | Hernandez | william@example.com | 123-456-7890 | 123 Elm St    |              2 |
|         11 | Emily     | Taylor    | emily.t@example.com | 987-654-3210 | 789 Maple St  |           NULL |
+------------+-----------+-----------+---------------------+--------------+---------------+----------------+
```

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, Customers.FirstName, Customers.LastName, Orders.OrderDate, Orders.TotalAmount
    -> FROM Orders
    -> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
+---------+-----------+-----------+------------+-------------+
| OrderID | FirstName | LastName  | OrderDate  | TotalAmount |
+---------+-----------+-----------+------------+-------------+
|     101 | John      | Doe       | 2024-04-01 |        1099 |
|     102 | Jane      | Smith     | 2024-04-02 |        2307 |
|     103 | sai       | kosh      | 2024-04-03 |         658 |
|     104 | Bob       | Williams  | 2024-04-04 |        1095 |
|     105 | Emily     | Brown     | 2024-04-05 |         328 |
|     106 | Michael   | Jones     | 2024-04-06 |         879 |
|     107 | Sarah     | Garcia    | 2024-04-07 |       10003 |
|     108 | David     | Martinez  | 2024-04-08 |         261 |
|     109 | Jennifer  | Rodriguez | 2024-04-09 |        3294 |
|     110 | William   | Hernandez | 2024-04-10 |         284 |
|     111 | Emily     | Taylor    | 2024-04-11 |        NULL |
+---------+-----------+-----------+------------+-------------+
```

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 productname,totalamount
    -> from products
    -> join orders on products.productid=orders.customerid;
+----------------+-------------+
| productname    | totalamount |
+----------------+-------------+
| Laptop         |        1099 |
| Smartphone     |        2307 |
| Tablet         |         658 |
| Smartwatch     |        1095 |
| Headphones     |         328 |
| Camera         |         879 |
| TV             |       10003 |
| Speaker        |         261 |
| Gaming Console |        3294 |
| Router         |         284 |
| Smart Speaker  |        NULL |
+----------------+-------------+
```

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

```
mysql> select concat(firstname,' ',lastname) as CustomerName,quantity
    -> From customers as c
    -> join orderdetails as O on c.customerid=O.productid
    -> where quantity>=1;
+------------------+----------+
| CustomerName     | quantity |
+------------------+----------+
| John Doe         |        1 |
| Jane Smith       |        3 |
| sai kosh         |        2 |
| Emily Brown      |        2 |
| Michael Jones    |        1 |
| Sarah Garcia     |        7 |
| David Martinez   |        3 |
| William Hernandez|        2 |
+------------------+----------+
```

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,quantity
    -> FROM Products p
    -> JOIN OrderDetails od ON p.ProductID = od.ProductID
    -> ORDER BY od.Quantity DESC
    -> LIMIT 1;
+-------------+----------+
| ProductName | quantity |
+-------------+----------+
| TV          |        7 |
+-------------+----------+
```

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

```
mysql> select Productname,Description from products;
+----------------+-----------------------------------------+
| Productname    | Description                             |
+----------------+-----------------------------------------+
| Laptop         | High-performance laptop with SSD        |
| Smartphone     | Latest model with dual camera           |
| Tablet         | 10-inch tablet with touchscreen         |
| Smartwatch     | Fitness tracker with heart rate monitor |
| Headphones     | Noise-canceling wireless headphones     |
| Camera         | DSLR camera with 18-55mm lens           |
| TV             | 4K Ultra HD smart TV                    |
| Speaker        | Bluetooth portable speaker              |
| Gaming Console | Next-gen gaming console                 |
| Router         | High-speed Wi-Fi router                 |
| Smart Speaker  | Voice-controlled smart speaker          |
+----------------+-----------------------------------------+
```

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.firstname,c.lastname,avg(o.totalamount)as AverageOrdervalue
    -> From Customers c Join orders o on c.customerid=o.customerid
    -> Group by c.customerid,c.firstname,c.lastname;
+-----------+-----------+-------------------+
| firstname | lastname  | AverageOrdervalue |
+-----------+-----------+-------------------+
| John      | Doe       |         1099.0000 |
| Jane      | Smith     |         2307.0000 |
| Bob       | Williams  |         1095.0000 |
| Emily     | Brown     |          328.0000 |
| Michael   | Jones     |          879.0000 |
| Sarah     | Garcia    |        10003.0000 |
| David     | Martinez  |          261.0000 |
| Jennifer  | Rodriguez |         3294.0000 |
| Emily     | Taylor    |              NULL |
+-----------+-----------+-------------------+
```

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 concat(firstname,' ',lastname) as CustomerName,orderid,totalamount
    -> from customers
    -> join orders on orders.customerid=customers.customerid
    -> order by totalamount desc limit 1;
+--------------+---------+-------------+
| CustomerName | orderid | totalamount |
+--------------+---------+-------------+
| Sarah Garcia |     107 |       10003 |
+--------------+---------+-------------+
```

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

```
mysql>  select productname,count(productname) from products group by productname;
+---------------+--------------------+
| productname   | count(productname) |
+---------------+--------------------+
| Laptop        |                  1 |
| Camera        |                  3 |
| Tablet        |                  1 |
| Headphones    |                  1 |
| TV            |                  1 |
| Speaker       |                  1 |
| Gaming Console |                 1 |
| Router        |                  1 |
| Smart Speaker |                  1 |
+---------------+--------------------+
```

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 concat(firstname,' ',lastname) as CustomerName
    -> from customers
    -> join products on customers.customerid=products.productid
    -> where productname='Camera';
+---------------+
| CustomerName  |
+---------------+
| Jane Smith    |
| Bob Williams  |
| Michael Jones |
+---------------+
```

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> set @startdate='2024-04-02',@enddate='2024-04-07';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT SUM(TotalAmount) AS TotalRevenue
    -> FROM Orders
    -> WHERE OrderDate BETWEEN @startdate AND @enddate;
+--------------+
| TotalRevenue |
+--------------+
|        14612 |
+--------------+
```

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);
+------------+-----------+----------+--------------------+--------------+------------+
| customerid | firstname | lastname | email              | phone        | address    |
+------------+-----------+----------+--------------------+--------------+------------+
|         12 | ranjith   | srinivas | ranjith@example.com | 354-364-3223 | 678 Min St |
+------------+-----------+----------+--------------------+--------------+------------+
```

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

```
mysql> SELECT COUNT(*) AS TotalProducts
    -> FROM Products;
+-----------------+
| TotalProducts   |
+-----------------+
|              11 |
+-----------------+
```

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

```
mysql> SELECT SUM(TotalAmount) AS TotalRevenue
    -> FROM Orders;
+--------------+
| TotalRevenue |
+--------------+
|        20208 |
+--------------+
```

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 productname,avg(od.quantity) as averagequantityordered
    -> from orderdetails od
    -> join products p on od.productid=p.productid
    -> where p.productname=@productname
    -> group by p.productname;
+-------------+-----------------------+
| productname | averagequantityordered |
+-------------+-----------------------+
| Laptop      |                1.0000 |
+-------------+-----------------------+
```

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 customerid,totalamount as TotalRevenue
    -> from orders
    -> where customerid=9;
+------------+--------------+
| customerid | TotalRevenue |
+------------+--------------+
|          9 |         3294 |
+------------+--------------+
```

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 concat(firstname,' ',lastname) as CustomerName,quantity as NumberofOrders
    -> from customers
    -> join orderdetails on customers.customerid=orderdetails.productid
    -> order by quantity desc limit 1;
+--------------+---------------+
| CustomerName | NumberofOrders |
+--------------+---------------+
| Sarah Garcia |             7 |
+--------------+---------------+
```

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,quantity
    -> FROM Products p
    -> JOIN OrderDetails od ON p.ProductID = od.ProductID
    -> ORDER BY od.Quantity DESC
    -> LIMIT 1;
+-------------+----------+
| ProductName | quantity |
+-------------+----------+
| TV          |        7 |
+-------------+----------+
```

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 concat(firstname,' ',lastname) as Customername,TotalAmount
    -> From Customers as C
    -> Join orders as O on C.customerid=O.customerid
    -> order by TotalAmount desc Limit 1;
+-------------+-------------+
| Customername | TotalAmount |
+-------------+-------------+
| Sarah Garcia |       10003 |
+-------------+-------------+
```

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 o;
+--------------------+
| AverageOrderValue |
+--------------------+
|          2020.8000 |
+--------------------+
```

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
    ->     FirstName,
    ->     LastName,
    ->     (
    ->         SELECT COUNT(*)
    ->         FROM Orders
    ->         WHERE Orders.CustomerID = Customers.CustomerID
    ->     ) AS OrderCount
    -> FROM
    ->     Customers;
+-----------+-----------+------------+
| FirstName | LastName  | OrderCount |
+-----------+-----------+------------+
| John      | Doe       |          1 |
| Jane      | Smith     |          1 |
| sai       | kosh      |          1 |
| Bob       | Williams  |          1 |
| Emily     | Brown     |          1 |
| Michael   | Jones     |          1 |
| Sarah     | Garcia    |          1 |
| David     | Martinez  |          1 |
| Jennifer  | Rodriguez |          1 |
| William   | Hernandez |          1 |
| Emily     | Taylor    |          1 |
| ranjith   | srinivas  |          0 |
+-----------+-----------+------------+
```