Name: Dheeraj Vemula
Assignment-1 (SQL)

Task 1:

1. Creating the database named “Techshop”.

```
mysql> CREATE DATABASE TechShop;
Query OK, 1 row affected (0.02 sec)

mysql> Use Techshop;
Database changed
```

2. Creating Tables for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema and appropriate Primary Key and Foreign Key constraints for referential integrity.

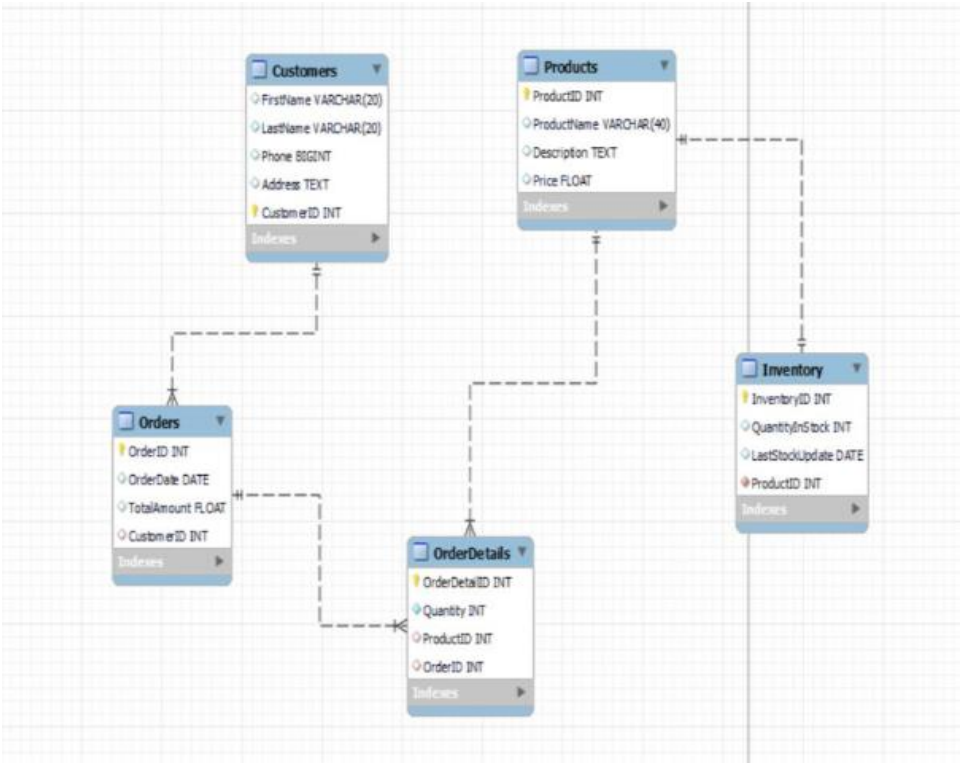
```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Description TEXT,
    Price DECIMAL(10,2)
);
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
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)
);
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate DATE,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

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

```
mysql> desc customers;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| CustomerID | int          | NO   | PRI | NULL    |       |
| FirstName  | varchar(50)  | YES  |     | NULL    |       |
| LastName   | varchar(50)  | YES  |     | NULL    |       |
| Email      | varchar(100) | YES  |     | NULL    |       |
| Phone      | varchar(20)  | YES  |     | NULL    |       |
| Address    | varchar(255) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
6 rows in set (0.04 sec)

mysql> desc inventory;
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| InventoryID     | int  | NO   | PRI | NULL    |       |
| ProductID       | int  | YES  | MUL | NULL    |       |
| QuantityInStock | int  | YES  |     | NULL    |       |
| LastStockUpdate | date | YES  |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
4 rows in set (0.01 sec)

mysql> desc orderdetails;
+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| OrderDetailID | int  | NO   | PRI | NULL    |       |
| OrderID       | int  | YES  | MUL | NULL    |       |
| ProductID     | int  | YES  | MUL | NULL    |       |
| Quantity      | int  | YES  |     | NULL    |       |
+---------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> desc orders;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| OrderID     | int           | NO   | PRI | NULL    |       |
| CustomerID  | int           | YES  | MUL | NULL    |       |
| OrderDate   | date          | YES  |     | NULL    |       |
| TotalAmount | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> desc products;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| ProductID   | int          | NO   | PRI | NULL    |       |
| ProductName | varchar(100) | YES  |     | NULL    |       |
| Description | text         | YES  |     | NULL    |       |
| Price       | decimal(10,2)| YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

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



5. Inserting 10 sample records into each of the following tables.

```
mysql> insert into customers(CustomerID, FirstName, LastName ,Email , Phone, Address)
    -> values
    ->
    -> (124, 'Jane', 'Smith', 'jane.smith@email.com', '555-5678', '456 Oak St'),
    -> (125, 'Alex', 'Johnson', 'alex.j@email.com', '555-9876', '789 Pine St'),
    -> (126, 'Sarah', 'Brown', 'sarah.b@email.com', '555-4321', '101 Elm St'),
    -> (127, 'Michael', 'Davis', 'michael.d@email.com', '555-8765', '202 Birch St'),
    -> (128, 'Emily', 'White', 'emily.w@email.com', '555-2345', '303 Maple St'),
    -> (129, 'David', 'Miller', 'david.m@email.com', '555-6789', '404 Cedar St'),
    -> (130, 'Jessica', 'Turner', 'jessica.t@email.com', '555-3456', '505 Walnut St'),
    -> (131, 'Brian', 'Clark', 'brian.c@email.com', '555-7890', '606 Fir St'),
    -> (132, 'Amanda', 'Lee', 'amanda.l@email.com', '555-5432', '707 Spruce St')
    -> ;
```

```
mysql> insert into products(ProductID, ProductName, Description, Price)
    -> values
    -> (1, 'Laptop', 'High-performance laptop', 999.99);
Query OK, 1 row affected (0.02 sec)

mysql> insert into products(ProductID, ProductName, Description, Price)
    -> values
    -> (2, 'Smartphone', 'Latest model smartphone', 699.99),
    -> (3, 'Headphones', 'Noise-canceling headphones', 129.99),
    -> (4, 'Digital Camera', '20MP digital camera', 449.99),
    -> (5, 'Smart Watch', 'Fitness tracking watch', 179.99),
    -> (6, 'Blender', 'High-power blender', 79.99),
    -> (7, 'Gaming Console', 'Next-gen gaming console', 499.99),
    -> (8, 'Wireless Speaker', 'Bluetooth wireless speaker', 59.99),
    -> (9, 'Coffee Maker', 'Programmable coffee maker', 129.99),
    -> (10, 'External Hard Drive', '2TB external hard drive', 89.99);
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

```
mysql> insert into orders(OrderID, CustomerID, OrderDate, TotalAmount)
    -> values
    -> (1, 123, '2024-01-15', 150.00),
    -> (2, 124, '2024-01-16', 299.99),
    -> (3, 124, '2024-01-17', 450.50),
    -> (4, 125, '2024-01-18', 75.99),
    -> (5, 126, '2024-01-19', 200.00),
    -> (6, 127, '2024-01-20', 1200.75),
    -> (7, 129, '2024-01-21', 50.00),
    -> (8, 130, '2024-01-22', 789.95),
    -> (9, 131, '2024-01-23', 600.50),
    -> (10, 132, '2024-01-24', 99.99);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> insert into orderdetails(OrderDetailID, OrderID, ProductID, Quantity)
    -> values
    -> (1, 1, 1, 2),
    -> (2, 1, 2, 1),
    -> (3, 2, 5, 3),
    -> (4, 3, 3, 1),
    -> (5, 3, 7, 2),
    -> (6, 4, 4, 1),
    -> (7, 5, 8, 4),
    -> (8, 5, 9, 1),
    -> (9, 6, 9, 2),
    -> (10, 7, 10, 1);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> insert into inventory(InventoryID, ProductID, QuantityInStock, LastStockUpdate)
    -> values
    -> (1, 1, 50, '2024-01-15'),
    -> (2, 2, 30, '2024-01-16'),
    -> (3, 3, 100, '2024-01-17'),
    -> (4, 4, 25, '2024-01-18'),
    -> (5, 5, 80, '2024-01-19'),
    -> (6, 6, 10, '2024-01-20'),
    -> (7, 7, 60, '2024-01-21'),
    -> (8, 8, 45, '2024-01-22'),
    -> (9, 9, 75, '2024-01-23'),
    -> (10, 10, 15, '2024-01-24');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

Task 2:

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@email.com   |
| Jane      | Smith    | jane.smith@email.com |
| Alex      | Johnson  | alex.j@email.com     |
| Sarah     | Brown    | sarah.b@email.com    |
| Michael   | Davis    | michael.d@email.com  |
| Emily     | White    | emily.w@email.com    |
| David     | Miller   | david.m@email.com    |
| Jessica   | Turner   | jessica.t@email.com  |
| Brian     | Clark    | brian.c@email.com    |
| Amanda    | Lee      | amanda.l@email.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 = customers.CustomerID;
+---------+------------+-----------+----------+
| OrderID | OrderDate  | FirstName | LastName |
+---------+------------+-----------+----------+
|       1 | 2024-01-15 | John      | Doe      |
|       2 | 2024-01-16 | Jane      | Smith    |
|       3 | 2024-01-17 | Jane      | Smith    |
|       4 | 2024-01-18 | Alex      | Johnson  |
|       5 | 2024-01-19 | Sarah     | Brown    |
|       6 | 2024-01-20 | Michael   | Davis    |
|       7 | 2024-01-21 | David     | Miller   |
|       8 | 2024-01-22 | Jessica   | Turner   |
|       9 | 2024-01-23 | Brian     | Clark    |
|      10 | 2024-01-24 | Amanda    | Lee      |
+---------+------------+-----------+----------+
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,'dheeru', 'bhai', 'dheerubhai@example.com', '555-1234', '925 Main St');
Query OK, 1 row affected (0.01 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
    -> WHERE Description = 'High-quality tablet';
Query OK, 2 rows affected, 2 warnings (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 2
```

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> INSERT INTO Orders (OrderID,CustomerID, OrderDate, TotalAmount)
    -> VALUES (11,3,'2023-07-01',1299.99);
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 values(501,1,'2024-01-12',1250.99);
Query OK, 1 row affected (0.01 sec)

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderId | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     501 |          1 | 2024-01-12 |     1250.99 |
|     502 |          2 | 2024-01-11 |      799.99 |
|     503 |          3 | 2024-01-12 |      219.95 |
|     504 |          4 | 2024-01-13 |      499.99 |
|     505 |          5 | 2024-01-14 |      199.99 |
|     506 |          6 | 2024-01-15 |      899.95 |
|     507 |          7 | 2024-01-16 |      129.99 |
|     508 |          8 | 2024-01-17 |       79.99 |
|     509 |          9 | 2024-01-18 |       29.99 |
|     510 |         10 | 2024-01-19 |      149.99 |
+---------+------------+------------+-------------+
10 rows in set (0.00 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 = 'new.email@email.com', Address = '456 Updated St'
    -> WHERE CustomerID = 1;
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 * Price)
    ->     FROM OrderDetails
    ->     JOIN Products ON OrderDetails.ProductID = Products.ProductID
    ->     WHERE OrderDetails.OrderID = Orders.OrderID
    -> )
    -> ;
Query OK, 11 rows affected (0.01 sec)
Rows matched: 11  Changed: 11  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 Cust
omerID = 3);
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM Orders WHERE CustomerID = 3;
Query OK, 2 rows 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,'Phone', 'Smart Phone', 499.99);
Query OK, 1 row affected (0.00 sec)
```

11. 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> select customerid,firstname,lastname,(select count(orderid) from orders where customers.customerid=orders.customerid) as NumberOfOrders
    -> from customers;
+------------+-----------+-----------+---------------+
| customerid | firstname | lastname  | NumberOfOrders |
+------------+-----------+-----------+---------------+
|          1 | John      | Doe       |             1 |
|          2 | Jane      | Smith     |             1 |
|          3 | Bob       | Johnson   |             1 |
|          4 | Alice     | Brown     |             1 |
|          5 | Charlie   | Davis     |             0 |
|          6 | Emily     | White     |             1 |
|          7 | David     | Miller    |             1 |
|          8 | Grace     | Wilson    |             1 |
|          9 | Henry     | Lee       |             1 |
|         10 | Olivia    | Turner    |             1 |
|         11 | Sarthak   | Shandilya |             0 |
+------------+-----------+-----------+---------------+
11 rows in set (0.01 sec)
```

Task 3:

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

```
SELECT Orders.OrderID, OrderDate, CONCAT(FirstName, ' ', LastName) A' at line 1
mysql> SELECT Orders.OrderID, OrderDate, CONCAT(FirstName, ' ', LastName) AS CustomerNa
me
    -> FROM Orders
    -> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
+---------+------------+----------------------+
| OrderID | OrderDate  | CustomerName         |
+---------+------------+----------------------+
|       1 | 2023-01-01 | John Doe             |
|       2 | 2023-02-15 | Jane Smith           |
|       4 | 2023-04-20 | Priya Sundaram       |
|       5 | 2023-05-15 | Karthik Venkataraman |
|       6 | 2023-06-25 | Aishwarya Natarajan  |
|       7 | 2023-03-10 | Ganesh Iyer          |
|       8 | 2023-04-20 | Meera Srinivasan     |
|       9 | 2023-05-15 | Suresh Rajagopal     |
|      10 | 2023-06-25 | Deepa Ganesan        |
+---------+------------+----------------------+
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 Products.ProductID, ProductName, SUM(Quantity * Price) AS TotalRevenue
    -> FROM OrderDetails
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE Products.Description = 'High-quality tablet'
    -> GROUP BY Products.ProductID, ProductName;
+-----------+-------------+--------------+
| ProductID | ProductName | TotalRevenue |
+-----------+-------------+--------------+
|         7 | Tablet      |      1099.98 |
+-----------+-------------+--------------+
1 row 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 DISTINCT Customers.CustomerID, FirstName, LastName, Email, Phone, Address

    -> FROM Customers
    -> JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
+------------+-----------+-------------+-------------------------+------------+------
------------------------+
| CustomerID | FirstName | LastName    | Email                   | Phone      | Addre
ss                     |
+------------+-----------+-------------+-------------------------+------------+------
------------------------+
|          1 | John      | Doe         | new.email@email.com     | 1234567890 | 456 U
pdated St              |
|          2 | Jane      | Smith       | jane.smith@email.com    | 9876543210 | 456 O
ak St                  |
|          4 | Priya     | Sundaram    | priya.sundaram@email.com | 2345678901 | 789 B
anana St, Hyderabad    |
|          5 | Karthik   | Venkataraman | karthik.venkat@email.com | 4567890123 | 890 M
ango St, Bangalore     |
|          6 | Aishwarya | Natarajan   | aishwarya.nat@email.com | 1232345678 | 123 P
ineapple St, Coimbatore |
|          7 | Ganesh    | Iyer        | ganesh.iyer@email.com   | 5678901234 | 234 P
apaya St, Mysuru       |
|          8 | Meera     | Srinivasan  | meera.srini@email.com   | 9012345678 | 345 G
uava St, Trivandrum    |
|          9 | Suresh    | Rajagopal   | suresh.raj@email.com    | 3456789012 | 456 A
pple St, Kochi         |
|         10 | Deepa     | Ganesan     | deepa.gan@email.com     | 6789012345 | 567 O
range St, Mangalore    |
+------------+-----------+-------------+-------------------------+------------+------
------------------------+
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 TOP 1 Products.ProductID, ProductName, SUM(Quantity) AS TotalQuantityOrde
red
    -> FROM OrderDetails
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE Products.Category = 'Electronic Gadgets'
    -> GROUP BY Products.ProductID, ProductName
    -> ORDER BY TotalQuantityOrdered DESC;
```

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

```
mysql> SELECT Orders.CustomerID, FirstName, LastName, AVG(TotalAmount) AS AverageOrderV
alue
    -> FROM Orders
    -> JOIN Customers ON Orders.CustomerID = Customers.CustomerID
    -> GROUP BY Orders.CustomerID, FirstName, LastName;
+------------+-----------+-------------+-------------------+
| CustomerID | FirstName | LastName    | AverageOrderValue |
+------------+-----------+-------------+-------------------+
|          1 | John      | Doe         |       2699.970000 |
|          2 | Jane      | Smith       |              NULL |
|          4 | Priya     | Sundaram    |        399.980000 |
|          5 | Karthik   | Venkataraman |       1299.990000 |
|          6 | Aishwarya | Natarajan   |        799.990000 |
|          7 | Ganesh    | Iyer        |       1099.980000 |
|          8 | Meera     | Srinivasan  |        199.990000 |
|          9 | Suresh    | Rajagopal   |       1299.990000 |
|         10 | Deepa     | Ganesan     |       1599.980000 |
+------------+-----------+-------------+-------------------+
9 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 Products.ProductID, ProductName, SUM(Quantity * Price) AS TotalRevenue
    -> FROM OrderDetails
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE Products.Description = 'High-performance laptop'
    -> GROUP BY Products.ProductID, ProductName;
+-----------+-------------+--------------+
| ProductID | ProductName | TotalRevenue |
+-----------+-------------+--------------+
|         1 | Laptop      |      1999.98 |
+-----------+-------------+--------------+
1 row 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 Products.ProductID, ProductName, COUNT(OrderDetails.OrderID) AS OrderCoun
t
    -> FROM Products
    -> LEFT JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID
    -> WHERE Products.Description = 'High-quality tablet'
    -> GROUP BY Products.ProductID, ProductName;
+-----------+-------------+------------+
| ProductID | ProductName | OrderCount |
+-----------+-------------+------------+
|         3 | Tablet      |          0 |
|         7 | Tablet      |          1 |
+-----------+-------------+------------+
2 rows 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 TOP 1 OrderID, OrderDate, CONCAT(FirstName, ' ', LastName) AS CustomerNam
e, TotalAmount
    -> FROM Orders
    -> JOIN Customers ON Orders.CustomerID = Customers.CustomerID
    -> ORDER BY TotalAmount DESC;
```

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 DISTINCT Customers.CustomerID, FirstName, LastName, Email, Phone, Address

    -> FROM Customers
    -> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE Products.ProductName = 'Laptop';
+------------+-----------+----------+--------------------+------------+--------------
-+
| CustomerID | FirstName | LastName | Email              | Phone      | Address
 |
+------------+-----------+----------+--------------------+------------+--------------
-+
|          1 | John      | Doe      | new.email@email.com | 1234567890 | 456 Updated St
 |
+------------+-----------+----------+--------------------+------------+--------------
-+
1 row in set (0.00 sec)
```

Task 4:

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

```
mysql> SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> FROM Customers
    -> LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> WHERE Orders.OrderID IS NULL;
+------------+-----------+----------+
| CustomerID | FirstName | LastName |
+------------+-----------+----------+
|          3 | Rajesh    | Kumar    |
|         11 | Anusha    | Chavva   |
+------------+-----------+----------+
2 rows in set (0.03 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 |
+--------------+
|      9399.87 |
+--------------+
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(Quantity) AS AverageQuantity
    -> FROM OrderDetails
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE Products.Description = 'Electronics';
+-----------------+
| AverageQuantity |
+-----------------+
|            NULL |
+-----------------+
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 = 1;
+--------------+
| TotalRevenue |
+--------------+
|      2699.97 |
+--------------+
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 TOP 1 Customers.FirstName, Customers.LastName, COUNT(Orders.OrderID) AS O
rderCount
    -> FROM Customers
    -> LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> ORDER BY OrderCount DESC;
```

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 TOP 1 Products.CategoryName, SUM(OrderDetails.Quantity) AS TotalQuantityO
rdered
    -> FROM OrderDetails
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> GROUP BY Products.CategoryName
    -> ORDER BY TotalQuantityOrdered DESC;
```

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 TOP 1 Customers.FirstName, Customers.LastName, SUM(OrderDetails.Quantity
* Products.Price) AS TotalSpending
    -> FROM Customers
    -> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE Products.CategoryName = 'Electronics'
    -> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> ORDER BY TotalSpending DESC;
```

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(TotalAmount) AS AverageOrderValue
    -> FROM Orders;
+-------------------+
| AverageOrderValue |
+-------------------+
|       1174.983750 |
+-------------------+
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 Customers.FirstName, Customers.LastName, COUNT(Orders.OrderID) AS OrderCo
unt
    -> FROM Customers
    -> LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName;
+-----------+-------------+------------+
| FirstName | LastName    | OrderCount |
+-----------+-------------+------------+
| John      | Doe         |          1 |
| Jane      | Smith       |          1 |
| Rajesh    | Kumar       |          0 |
| Priya     | Sundaram    |          1 |
| Karthik   | Venkataraman |          1 |
| Aishwarya | Natarajan   |          1 |
| Ganesh    | Iyer        |          1 |
| Meera     | Srinivasan  |          1 |
| Suresh    | Rajagopal   |          1 |
| Deepa     | Ganesan     |          1 |
| Anusha    | Chavva      |          0 |
+-----------+-------------+------------+
11 rows in set (0.01 sec)
```