Task:1. Database Design:

1. Create the database named "TechShop"

MySQL 8.0 Command Line Client

```
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database TechShop;
Query OK, 1 row affected (0.00 sec)

mysql> use TechShop;
Database changed
```

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

```
mysql> use TechShop;
Database changed
mysql> CREATE TABLE Customers(
    -> CustomerID INT PRIMARY KEY,
    -> FirstName VARCHAR(255),
    -> LastName VARCHAR(255),
    -> Email VARCHAR(255),
    -> Phone VARCHAR(20),
    -> Address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> desc Customers;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| CustomerID | int         | NO   | PRI | NULL    |       |
| FirstName  | varchar(255) | YES  |     | NULL    |       |
| LastName   | varchar(255) | YES  |     | NULL    |       |
| Email      | varchar(255) | YES  |     | NULL    |       |
| Phone      | varchar(20)  | YES  |     | NULL    |       |
| Address    | varchar(255) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> create table Products(
    -> ProductID int primary key,
    -> ProductName varchar(255),
    -> Description varchar(255),
    -> Price decimal(10,2)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> desc Products;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| ProductID   | int           | NO   | PRI | NULL    |       |
| ProductName | varchar(255)  | YES  |     | NULL    |       |
| Description | varchar(255)  | YES  |     | NULL    |       |
| Price       | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> create table Orders(
    -> OrderID int primary key,
    -> CustomerID int,
    -> OrderDate date,
    -> TotalAmount decimal(10,2),
    -> foreign key(CustomerID) references Customers(CustomerID)
    -> );
Query OK, 0 rows affected (0.06 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> create table OrderDetails(
    -> OrderDetailID int primary key,
    -> OrderID int,
    -> ProductID int,
    -> Quantity varchar(255),
    -> foreign key(OrderID) references Orders(OrderID),
    -> foreign key(ProductID) references Products(ProductID)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

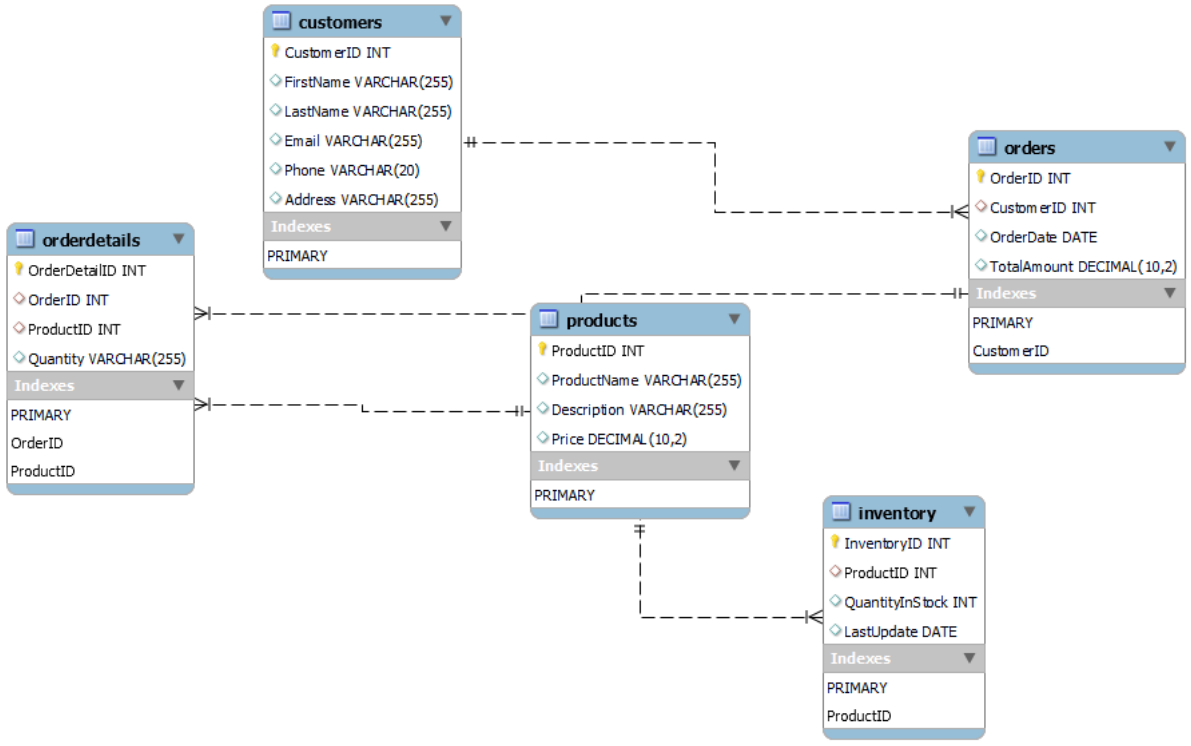
```
mysql> desc OrderDetails;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| OrderDetailID | int          | NO   | PRI | NULL    |       |
| OrderID       | int          | YES  | MUL | NULL    |       |
| ProductID     | int          | YES  | MUL | NULL    |       |
| Quantity      | varchar(255) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

MySQL 8.0 Command Line Client

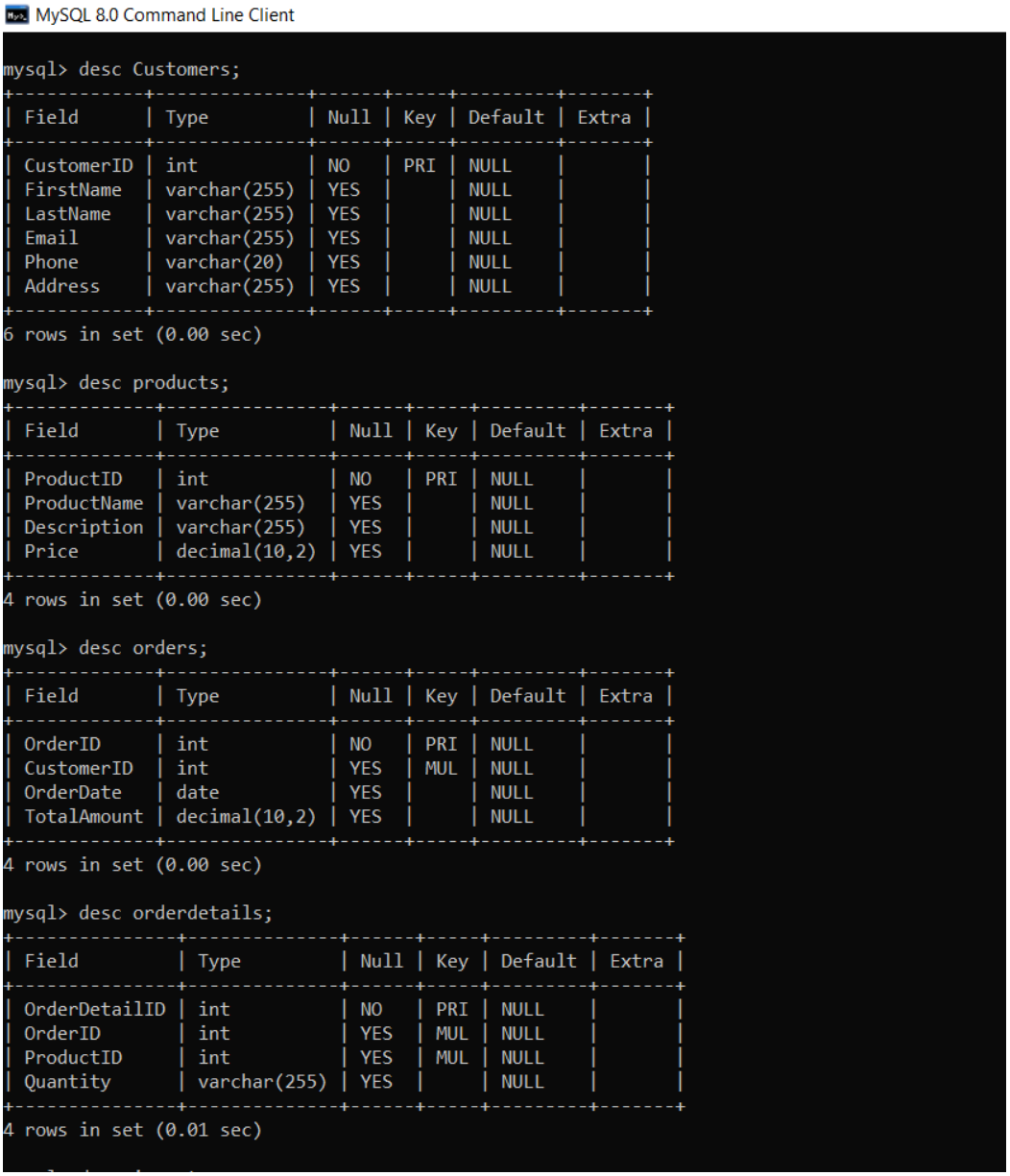
```
mysql> desc inventory;
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| InventoryID     | int  | NO   | PRI | NULL    |       |
| ProductID       | int  | YES  | MUL | NULL    |       |
| QuantityInStock | int  | YES  |     | NULL    |       |
| LastUpdate      | date | YES  |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
4 rows in set (0.02 sec)

mysql>
```

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



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




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

a. Customers

b. Products

c. Orders

d. OrderDetails

e. Inventory

MySQL 8.0 Command Line Client

```
mysql> insert into Customers(CustomerID, FirstName, LastName, Email, Phone, Address)
    -> values(1, 'Aman', 'Kumar', 'aman@gmail.com', '1234567897', 'Motihar'),
    -> (2, 'Ansu', 'Singh', 'anshu@gmail.com','1111111111', 'Bihar'),
    -> (3, 'Amit', 'Sah', 'amit@gmail.com','1111111112', 'Rachi'),
    -> (4, 'Asmita', 'Si', 'asmita@gmail.com','1111111121', 'Durgapur'),
    -> (5, 'Swati', 'Singh', 'swati@gmail.com','1111111122', 'Asansol'),
    -> (6, 'Prateeti', 'Maji', 'prateeti@gmail.com','1111111133', 'Durgapur'),
    -> (7, 'Anjali', 'Sneha', 'anjali@gmail.com','1111111134', 'Raghunathpur'),
    -> (8, 'Om', 'Kumar', 'om@gmail.com','1111111136', 'Gujrat'),
    -> (9, 'Roshan', 'Singh', 'roshan@gmail.com','1111111137', 'Gandhinagar'),
    -> (10, 'Arava', 'Krishnavenni', 'arava@gmail.com','1111111139', 'Chennai');
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Customers;
+------------+-----------+--------------+--------------------+------------+--------------+
| CustomerID | FirstName | LastName     | Email              | Phone      | Address      |
+------------+-----------+--------------+--------------------+------------+--------------+
|          1 | Aman      | Kumar        | aman@gmail.com     | 1234567897 | Motihar      |
|          2 | Ansu      | Singh        | anshu@gmail.com    | 1111111111 | Bihar        |
|          3 | Amit      | Sah          | amit@gmail.com     | 1111111112 | Rachi        |
|          4 | Asmita    | Si           | asmita@gmail.com   | 1111111121 | Durgapur     |
|          5 | Swati     | Singh        | swati@gmail.com    | 1111111122 | Asansol      |
|          6 | Prateeti  | Maji         | prateeti@gmail.com | 1111111133 | Durgapur     |
|          7 | Anjali    | Sneha        | anjali@gmail.com   | 1111111134 | Raghunathpur |
|          8 | Om        | Kumar        | om@gmail.com       | 1111111136 | Gujrat       |
|          9 | Roshan    | Singh        | roshan@gmail.com   | 1111111137 | Gandhinagar  |
|         10 | Arava     | Krishnavenni | arava@gmail.com    | 1111111139 | Chennai      |
+------------+-----------+--------------+--------------------+------------+--------------+
10 rows in set (0.00 sec)
```

```
mysql> select * from products;
+-----------+-------------+-------------------+-----------+
| ProductID | ProductName | Description       | Price     |
+-----------+-------------+-------------------+-----------+
|        11 | Laptop      | electronic gadget | 500000.14 |
|        12 | Mouse       | electronic gadget |    500.24 |
|        13 | Keyboard    | electronic gadget |    800.54 |
|        14 | Printer     | electronic gadget |   8000.34 |
|        15 | Phone       | electronic gadget |  18000.84 |
|        16 | WallClock   | Home appliances   |   5000.14 |
|        17 | Headphone   | electronic gadget |   2000.14 |
|        18 | Camera      | electronic gadget | 200000.99 |
|        19 | Smartwatch  | electronic gadget |   2000.55 |
|        20 | Speaker     | Home appliances   |  20000.85 |
+-----------+-------------+-------------------+-----------+
10 rows in set (0.00 sec)

mysql>
```

```
mysql> insert into Orders(OrderID, CustomerID, OrderDate, TotalAmount)
    -> values(101, 1, '2024-01-01', 500000.14),
    -> (102, 2, '2024-01-02', 500.24),
    -> (103, 3, '2024-01-03', 800.54),
    -> (104, 4, '2024-01-04', 8000.34),
    -> (105, 5, '2024-01-05', 18000.84),
    -> (106, 6, '2024-01-06', 5000.14),
    -> (107, 7, '2024-01-07', 2000.14),
    -> (108, 8, '2024-01-08', 200000.99),
    -> (109, 9, '2024-01-09', 2000.55),
    -> (110, 10, '2024-01-10', 20000.85);
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     101 |          1 | 2024-01-01 |   500000.14 |
|     102 |          2 | 2024-01-02 |      500.24 |
|     103 |          3 | 2024-01-03 |      800.54 |
|     104 |          4 | 2024-01-04 |     8000.34 |
|     105 |          5 | 2024-01-05 |    18000.84 |
|     106 |          6 | 2024-01-06 |     5000.14 |
|     107 |          7 | 2024-01-07 |     2000.14 |
|     108 |          8 | 2024-01-08 |   200000.99 |
|     109 |          9 | 2024-01-09 |     2000.55 |
|     110 |         10 | 2024-01-10 |    20000.85 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

```
mysql> insert into OrderDetails(OrderDetailID, OrderID, ProductID, Quantity)
    -> values(21,101,11,1),
    -> (22,102,12,2),
    -> (23,103,13,3),
    -> (24,104,14,4),
    -> (25,105,15,5),
    -> (26,106,16,6),
    -> (27,107,17,7),
    -> (28,108,18,8),
    -> (29,109,19,9),
    -> (30,110,20,10);
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|            21 |     101 |        11 | 1        |
|            22 |     102 |        12 | 2        |
|            23 |     103 |        13 | 3        |
|            24 |     104 |        14 | 4        |
|            25 |     105 |        15 | 5        |
|            26 |     106 |        16 | 6        |
|            27 |     107 |        17 | 7        |
|            28 |     108 |        18 | 8        |
|            29 |     109 |        19 | 9        |
|            30 |     110 |        20 | 10       |
+---------------+---------+-----------+----------+
10 rows in set (0.00 sec)
```

```
mysql> insert into Inventory(InventoryID, ProductID, QuantityInStock, LastUpdate)
    -> values(31, 11, 1, '2024-01-01'),
    -> (32, 12, 2, '2024-01-02'),
    -> (33, 13, 3, '2024-01-03'),
    -> (34, 14, 4, '2024-01-04'),
    -> (35, 15, 5, '2024-01-05'),
    -> (36, 16, 6, '2024-01-06'),
    -> (37, 17, 7, '2024-01-07'),
    -> (38, 18, 8, '2024-01-08'),
    -> (39, 19, 9, '2024-01-09'),
    -> (40, 20, 10, '2024-01-10');
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Inventory;
+-------------+-----------+-----------------+------------+
| InventoryID | ProductID | QuantityInStock | LastUpdate |
+-------------+-----------+-----------------+------------+
|          31 |        11 |               1 | 2024-01-01 |
|          32 |        12 |               2 | 2024-01-02 |
|          33 |        13 |               3 | 2024-01-03 |
|          34 |        14 |               4 | 2024-01-04 |
|          35 |        15 |               5 | 2024-01-05 |
|          36 |        16 |               6 | 2024-01-06 |
|          37 |        17 |               7 | 2024-01-07 |
|          38 |        18 |               8 | 2024-01-08 |
|          39 |        19 |               9 | 2024-01-09 |
|          40 |        20 |              10 | 2024-01-10 |
+-------------+-----------+-----------------+------------+
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 concat(FirstName, ' ' , LastName) as Name, Email from Customers;
+--------------------+--------------------+
| Name               | Email              |
+--------------------+--------------------+
| Aman Kumar         | aman@gmail.com     |
| Ansu Singh         | anshu@gmail.com    |
| Amit Sah           | amit@gmail.com     |
| Asmita Si          | asmita@gmail.com   |
| Swati Singh        | swati@gmail.com    |
| Prateeti Maji      | prateeti@gmail.com |
| Anjali Sneha       | anjali@gmail.com   |
| Om Kumar           | om@gmail.com       |
| Roshan Singh       | roshan@gmail.com   |
| Arava Krishnavenni | arava@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 orders.orderID, orders.orderDate, customers.firstname,
    -> customers.lastname from orders,
    -> customers where orders.customerID = customers.customerID;
+---------+------------+-----------+--------------+
| orderID | orderDate  | firstname | lastname     |
+---------+------------+-----------+--------------+
|     101 | 2024-01-01 | Aman      | Kumar        |
|     102 | 2024-01-02 | Ansu      | Singh        |
|     103 | 2024-01-03 | Amit      | Sah          |
|     104 | 2024-01-04 | Asmita    | Si           |
|     105 | 2024-01-05 | Swati     | Singh        |
|     106 | 2024-01-06 | Prateeti  | Maji         |
|     107 | 2024-01-07 | Anjali    | Sneha        |
|     108 | 2024-01-08 | Om        | Kumar        |
|     109 | 2024-01-09 | Roshan    | Singh        |
|     110 | 2024-01-10 | Arava     | Krishnavenni |
+---------+------------+-----------+--------------+
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,'Jasmine','Roy','jas123@gmail.com',2258746984,'Mumbai');
Query OK, 1 row affected (0.00 sec)

mysql> select * from customers;
+------------+-----------+--------------+--------------------+------------+--------------+
| CustomerID | FirstName | LastName     | Email              | Phone      | Address      |
+------------+-----------+--------------+--------------------+------------+--------------+
|          1 | Aman      | Kumar        | aman@gmail.com     | 1234567897 | Motihar      |
|          2 | Ansu      | Singh        | anshu@gmail.com    | 1111111111 | Bihar        |
|          3 | Amit      | Sah          | amit@gmail.com     | 1111111112 | Rachi        |
|          4 | Asmita    | Si           | asmita@gmail.com   | 1111111121 | Durgapur     |
|          5 | Swati     | Singh        | swati@gmail.com    | 1111111122 | Asansol      |
|          6 | Prateeti  | Maji         | prateeti@gmail.com | 1111111133 | Durgapur     |
|          7 | Anjali    | Sneha        | anjali@gmail.com   | 1111111134 | Raghunathpur |
|          8 | Om        | Kumar        | om@gmail.com       | 1111111136 | Gujrat       |
|          9 | Roshan    | Singh        | roshan@gmail.com   | 1111111137 | Gandhinagar  |
|         10 | Arava     | Krishnavenni | arava@gmail.com    | 1111111139 | Chennai      |
|         11 | Jasmine   | Roy          | jas123@gmail.com   | 2258746984 | Mumbai       |
+------------+-----------+--------------+--------------------+------------+--------------+
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 * 1.10
    -> WHERE Description = 'Electronic Gadget';
Query OK, 8 rows affected, 8 warnings (0.01 sec)
Rows matched: 8  Changed: 8  Warnings: 8

mysql> select * from Products;
+-----------+-------------+-------------------+-----------+
| ProductID | ProductName | Description       | Price     |
+-----------+-------------+-------------------+-----------+
|        11 | Laptop      | electronic gadget | 550000.15 |
|        12 | Mouse       | electronic gadget |    550.26 |
|        13 | Keyboard    | electronic gadget |    880.59 |
|        14 | Printer     | electronic gadget |   8800.37 |
|        15 | Phone       | electronic gadget |  19800.92 |
|        16 | WallClock   | Home appliances   |   5000.14 |
|        17 | Headphone   | electronic gadget |   2200.15 |
|        18 | Camera      | electronic gadget | 220001.09 |
|        19 | Smartwatch  | electronic gadget |   2200.61 |
|        20 | Speaker     | Home appliances   |  20000.85 |
+-----------+-------------+-------------------+-----------+
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> DELIMITER ##
mysql> CREATE PROCEDURE DeleteCustomerOrder(IN OrderIDParam INT)
    -> BEGIN
    ->     -- Delete from OrderDetails
    ->     DELETE FROM OrderDetails
    ->     WHERE OrderID = OrderIDParam;
    ->     -- Delete from Orders
    ->     DELETE FROM Orders WHERE OrderID = OrderIDParam;
    -> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call DeleteCustomerOrder(110);
Query OK, 1 row affected (0.02 sec)

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     101 |          1 | 2024-01-01 |   550000.15 |
|     102 |          2 | 2024-01-02 |     1100.52 |
|     103 |          3 | 2024-01-03 |     2641.77 |
|     104 |          4 | 2024-01-04 |    35201.48 |
|     105 |          5 | 2024-01-05 |    99004.60 |
|     106 |          6 | 2024-01-06 |    30000.84 |
|     107 |          7 | 2024-01-07 |    15401.05 |
|     108 |          8 | 2024-01-08 |  1760008.72 |
|     109 |          9 | 2024-01-09 |    19805.49 |
|     111 |          5 | 2024-01-11 |        NULL |
+---------+------------+------------+-------------+
10 rows in set (0.00 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, 5, '2024-01-11', 1550.00);
Query OK, 1 row affected (0.01 sec)

mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     101 |          1 | 2024-01-01 |   500000.14 |
|     102 |          2 | 2024-01-02 |      500.24 |
|     103 |          3 | 2024-01-03 |      800.54 |
|     104 |          4 | 2024-01-04 |     8000.34 |
|     105 |          5 | 2024-01-05 |    18000.84 |
|     106 |          6 | 2024-01-06 |     5000.14 |
|     107 |          7 | 2024-01-07 |     2000.14 |
|     108 |          8 | 2024-01-08 |   200000.99 |
|     109 |          9 | 2024-01-09 |     2000.55 |
|     110 |         10 | 2024-01-10 |    20000.85 |
|     111 |          5 | 2024-01-11 |     1550.00 |
+---------+------------+------------+-------------+
11 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> DELIMITER ##
mysql> create procedure updateInformation(IN CustomerIDParam INT, EmailParam VARCHAR(255), AddressParam VARCHAR(255))
    -> BEGIN
    -> UPDATE Customers
    -> SET email = EmailParam, Address = AddressParam
    -> WHERE CustomerID = CustomerIDParam;
    -> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call updateInformation(8,'om123@gmail.com','Bihar');
Query OK, 1 row affected (0.01 sec)

mysql> select * from Customers;
+------------+-----------+--------------+--------------------+------------+--------------+
| CustomerID | FirstName | LastName     | Email              | Phone      | Address      |
+------------+-----------+--------------+--------------------+------------+--------------+
|          1 | Aman      | Kumar        | aman@gmail.com     | 1234567897 | Motihar      |
|          2 | Ansu      | Singh        | anshu@gmail.com    | 1111111111 | Bihar        |
|          3 | Amit      | Sah          | amit@gmail.com     | 1111111112 | Rachi        |
|          4 | Asmita    | Si           | asmita@gmail.com   | 1111111121 | Durgapur     |
|          5 | Swati     | Singh        | swati@gmail.com    | 1111111122 | Asansol      |
|          6 | Prateeti  | Maji         | prateeti@gmail.com | 1111111133 | Durgapur     |
|          7 | Anjali    | Sneha        | anjali@gmail.com   | 1111111134 | Raghunathpur |
|          8 | Om        | Kumar        | om123@gmail.com    | 1111111136 | Bihar        |
|          9 | Roshan    | Singh        | roshan@gmail.com   | 1111111137 | Gandhinagar  |
|         10 | Arava     | Krishnavenni | arava@gmail.com    | 1111111139 | Chennai      |
|         11 | Jasmine   | Roy          | jas123@gmail.com   | 2258746984 | Mumbai       |
+------------+-----------+--------------+--------------------+------------+--------------+
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(Products.Price * OrderDetails.Quantity)
    -> from OrderDetails
    -> join products on OrderDetails.ProductID = Products.ProductID
    -> where OrderDetails.OrderID = Orders.OrderID)
    -> where OrderID in (select distinct OrderID from OrderDetails);
Query OK, 0 rows affected, 3 warnings (0.00 sec)
Rows matched: 9  Changed: 0  Warnings: 3

mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     101 |          1 | 2024-01-01 |   550000.15 |
|     102 |          2 | 2024-01-02 |     1100.52 |
|     103 |          3 | 2024-01-03 |     2641.77 |
|     104 |          4 | 2024-01-04 |    35201.48 |
|     105 |          5 | 2024-01-05 |    99004.60 |
|     106 |          6 | 2024-01-06 |    30000.84 |
|     107 |          7 | 2024-01-07 |    15401.05 |
|     108 |          8 | 2024-01-08 |  1760008.72 |
|     109 |          9 | 2024-01-09 |    19805.49 |
|     111 |          5 | 2024-01-11 |        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> DELIMITER ##
mysql> create procedure deleteOrders(IN CustomerIDParam INT)
    -> BEGIN
    -> DELETE FROM OrderDetails
    -> WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = CustomerIDParam);
    -> DELETE FROM Orders
    -> WHERE CustomerID = CustomerIDParam;
    -> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call deleteOrders(9);
Query OK, 1 row affected (0.01 sec)

mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     101 |          1 | 2024-01-01 |   550000.15 |
|     102 |          2 | 2024-01-02 |     1100.52 |
|     103 |          3 | 2024-01-03 |     2641.77 |
|     104 |          4 | 2024-01-04 |    35201.48 |
|     105 |          5 | 2024-01-05 |    99004.60 |
|     106 |          6 | 2024-01-06 |    30000.84 |
|     107 |          7 | 2024-01-07 |    15401.05 |
|     108 |          8 | 2024-01-08 |  1760008.72 |
|     111 |          5 | 2024-01-11 |        NULL |
+---------+------------+------------+-------------+
9 rows in set (0.00 sec)

mysql> select * from OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|            21 |     101 |        11 | 1        |
|            22 |     102 |        12 | 2        |
|            23 |     103 |        13 | 3        |
|            24 |     104 |        14 | 4        |
|            25 |     105 |        15 | 5        |
|            26 |     106 |        16 | 6        |
|            27 |     107 |        17 | 7        |
|            28 |     108 |        18 | 8        |
+---------------+---------+-----------+----------+
8 rows in set (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(21,'Monitor','electronic gadget', 8000.00);
Query OK, 1 row affected (0.00 sec)

mysql> select * from products;
+-----------+-------------+-------------------+-----------+
| ProductID | ProductName | Description       | Price     |
+-----------+-------------+-------------------+-----------+
|        11 | Laptop      | electronic gadget | 550000.15 |
|        12 | Mouse       | electronic gadget |    550.26 |
|        13 | Keyboard    | electronic gadget |    880.59 |
|        14 | Printer     | electronic gadget |   8800.37 |
|        15 | Phone       | electronic gadget |  19800.92 |
|        16 | WallClock   | Home appliances   |   5000.14 |
|        17 | Headphone   | electronic gadget |   2200.15 |
|        18 | Camera      | electronic gadget | 220001.09 |
|        19 | Smartwatch  | electronic gadget |   2200.61 |
|        20 | Speaker     | Home appliances   |  20000.85 |
|        21 | Monitor     | electronic gadget |   8000.00 |
+-----------+-------------+-------------------+-----------+
11 rows in set (0.00 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> ALTER TABLE Orders ADD COLUMN Status VARCHAR(50) DEFAULT 'Pending';
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DELIMITER ##
mysql> CREATE PROCEDURE UpdateOrderStatus(
    ->
    -> IN OrderIDParam INT,
    ->
    -> IN NewStatus VARCHAR(50)
    ->
    -> BEGIN
    ->
    -> UPDATE Orders
    ->
    -> SET Status = NewStatus
    ->
    -> WHERE OrderID = OrderIDParam;
    ->  END ##
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server

UPDATE Orders

SET Status = NewStatus

WHERE OrderID = OrderIDParam;
 END' at line 7
mysql>
mysql>  DELIMITER ;
mysql> DELIMITER ##
mysql> CREATE PROCEDURE UpdateOrdersStatus(
    -> IN OrderIDParam INT,
    ->
    -> IN NewStatus VARCHAR(50))
    ->
    -> BEGIN
    ->
    -> UPDATE Orders
    ->
    -> SET Status = NewStatus
    ->
    -> WHERE OrderID = OrderIDParam;
    ->  END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
```

```
mysql> CALL UpdateOrdersStatus(101, 'Shipped');
Query OK, 1 row affected (0.00 sec)

mysql> select * from Orders;
+---------+------------+------------+-------------+---------+
| OrderID | CustomerID | OrderDate  | TotalAmount | Status  |
+---------+------------+------------+-------------+---------+
|     101 |          1 | 2024-01-01 |   550000.15 | Shipped |
|     102 |          2 | 2024-01-02 |     1100.52 | Pending |
|     103 |          3 | 2024-01-03 |     2641.77 | Pending |
|     104 |          4 | 2024-01-04 |    35201.48 | Pending |
|     105 |          5 | 2024-01-05 |    99004.60 | Pending |
|     106 |          6 | 2024-01-06 |    30000.84 | Pending |
|     107 |          7 | 2024-01-07 |    15401.05 | Pending |
|     108 |          8 | 2024-01-08 |  1760008.72 | Pending |
|     111 |          5 | 2024-01-11 |        NULL | Pending |
+---------+------------+------------+-------------+---------+
9 rows in set (0.00 sec)
```

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 8.0 Command Line Client

```
mysql> alter table Customers
    -> add column NumberOfOrders int;
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> update Customers
    -> set NumberOfOrders = (
    -> select count(OrderID)
    -> from Orders
    -> where Customers.CustomerID = Orders.CustomerID
    -> );
Query OK, 11 rows affected (0.01 sec)
Rows matched: 11  Changed: 11  Warnings: 0

mysql> select * from Customers;
+------------+-----------+-------------+--------------------+------------+-------------+----------------+
| CustomerID | FirstName | LastName    | Email              | Phone      | Address     | NumberOfOrders |
+------------+-----------+-------------+--------------------+------------+-------------+----------------+
|          1 | Aman      | Kumar       | aman@gmail.com     | 1234567897 | Motihar     |              1 |
|          2 | Ansu      | Singh       | anshu@gmail.com    | 1111111111 | Bihar       |              1 |
|          3 | Amit      | Sah         | amit@gmail.com     | 1111111112 | Rachi       |              1 |
|          4 | Asmita    | Si          | asmita@gmail.com   | 1111111121 | Durgapur    |              1 |
|          5 | Swati     | Singh       | swati@gmail.com    | 1111111122 | Asansol     |              2 |
|          6 | Prateeti  | Maji        | prateeti@gmail.com | 1111111133 | Durgapur    |              1 |
|          7 | Anjali    | Sneha       | anjali@gmail.com   | 1111111134 | Raghunathpur |              1 |
|          8 | Om        | Kumar       | om123@gmail.com    | 1111111136 | Bihar       |              1 |
|          9 | Roshan    | Singh       | roshan@gmail.com   | 1111111137 | Gandhinagar |              0 |
|         10 | Arava     | Krishnavenni | arava@gmail.com    | 1111111139 | Chennai     |              0 |
|         11 | Jasmine   | Roy         | jas123@gmail.com   | 2258746984 | Mumbai      |              0 |
+------------+-----------+-------------+--------------------+------------+-------------+----------------+
11 rows in set (0.00 sec)
```

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 | Aman      | Kumar    | 2024-01-01 |   550000.15 |
|     102 | Ansu      | Singh    | 2024-01-02 |     1100.52 |
|     103 | Amit      | Sah      | 2024-01-03 |     2641.77 |
|     104 | Asmita    | Si       | 2024-01-04 |    35201.48 |
|     105 | Swati     | Singh    | 2024-01-05 |    99004.60 |
|     106 | Prateeti  | Maji     | 2024-01-06 |    30000.84 |
|     107 | Anjali    | Sneha    | 2024-01-07 |    15401.05 |
|     108 | Om        | Kumar    | 2024-01-08 |  1760008.72 |
|     111 | Swati     | Singh    | 2024-01-11 |        NULL |
+---------+-----------+----------+------------+-------------+
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, Products.ProductName,
    -> SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue
    -> FROM Products
    -> JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID
    -> GROUP BY Products.ProductID, Products.ProductName;
+-----------+-------------+--------------------+
| ProductID | ProductName | TotalRevenue       |
+-----------+-------------+--------------------+
|        11 | Laptop      |          550000.15 |
|        12 | Mouse       |            1100.52 |
|        13 | Keyboard    |            2641.77 |
|        14 | Printer     |           35201.48 |
|        15 | Phone       |  99004.59999999999 |
|        16 | WallClock   | 30000.840000000004 |
|        17 | Headphone   | 15401.050000000001 |
|        18 | Camera      |         1760008.72 |
+-----------+-------------+--------------------+
8 rows in set (0.00 sec)
```

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

```
mysql> SELECT DISTINCT c.CustomerID,c.FirstName,c.LastName,c.Email,
    -> c.Phone,c.Address FROM Customers c JOIN Orders ON c.CustomerID = Orders.CustomerID;
+------------+-----------+----------+--------------------+------------+--------------+
| CustomerID | FirstName | LastName | Email              | Phone      | Address      |
+------------+-----------+----------+--------------------+------------+--------------+
|          1 | Aman      | Kumar    | aman@gmail.com     | 1234567897 | Motihar      |
|          2 | Ansu      | Singh    | anshu@gmail.com    | 1111111111 | Bihar        |
|          3 | Amit      | Sah      | amit@gmail.com     | 1111111112 | Rachi        |
|          4 | Asmita    | Si       | asmita@gmail.com   | 1111111121 | Durgapur     |
|          5 | Swati     | Singh    | swati@gmail.com    | 1111111122 | Asansol      |
|          6 | Prateeti  | Maji     | prateeti@gmail.com | 1111111133 | Durgapur     |
|          7 | Anjali    | Sneha    | anjali@gmail.com   | 1111111134 | Raghunathpur |
|          8 | Om        | Kumar    | om123@gmail.com    | 1111111136 | Bihar        |
+------------+-----------+----------+--------------------+------------+--------------+
8 rows in set (0.00 sec)
```

4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest total quantity ordered. Include the product name and the total quantity ordered.

MySQL 8.0 Command Line Client

```
mysql> SELECT p.ProductID, p.ProductName, SUM(OrderDetails.Quantity) AS TotalQuantityOrdered
    -> FROM Products p JOIN OrderDetails ON p.ProductID = OrderDetails.ProductID GROUP BY p.ProductID, p.ProductName
    -> ORDER BY TotalQuantityOrdered DESC LIMIT 1;
+-----------+-------------+----------------------+
| ProductID | ProductName | TotalQuantityOrdered |
+-----------+-------------+----------------------+
|        18 | Camera      |                    8 |
+-----------+-------------+----------------------+
1 row in set (0.00 sec)

mysql>
```

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

MySQL 8.0 Command Line Client

```
mysql> select ProductID,ProductName, Description from products;
+-----------+-------------+-------------------+
| ProductID | ProductName | Description       |
+-----------+-------------+-------------------+
|        11 | Laptop      | electronic gadget |
|        12 | Mouse       | electronic gadget |
|        13 | Keyboard    | electronic gadget |
|        14 | Printer     | electronic gadget |
|        15 | Phone       | electronic gadget |
|        16 | WallClock   | Home appliances   |
|        17 | Headphone   | electronic gadget |
|        18 | Camera      | electronic gadget |
|        19 | Smartwatch  | electronic gadget |
|        20 | Speaker     | Home appliances   |
|        21 | Monitor     | electronic gadget |
+-----------+-------------+-------------------+
11 rows in set (0.00 sec)

mysql>
```

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

```
mysql> select c.CustomerID, c.FirstName, c.LastName , avg(Orders.TotalAmount) as AverageOrderValue
    -> from Customers c join Orders on c.CustomerID = Orders.CustomerID group by
    -> c.CustomerID, c.FirstName, c.LastName;
+------------+-----------+----------+-------------------+
| CustomerID | FirstName | LastName | AverageOrderValue |
+------------+-----------+----------+-------------------+
|          1 | Aman      | Kumar    |     550000.150000 |
|          2 | Ansu      | Singh    |       1100.520000 |
|          3 | Amit      | Sah      |       2641.770000 |
|          4 | Asmita    | Si       |      35201.480000 |
|          5 | Swati     | Singh    |      99004.600000 |
|          6 | Prateeti  | Maji     |      30000.840000 |
|          7 | Anjali    | Sneha    |      15401.050000 |
|          8 | Om        | Kumar    |    1760008.720000 |
+------------+-----------+----------+-------------------+
8 rows in set (0.01 sec)

mysql>
```

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

```
mysql> SELECT
    ->     o.OrderID,
    ->     c.CustomerID,
    ->     c.FirstName,
    ->     c.LastName,
    ->     c.Email,
    ->     c.Phone,
    ->     c.Address,
    ->     SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue
    -> FROM Orders o JOIN Customers c ON o.CustomerID = c.CustomerID
    -> JOIN OrderDetails ON o.OrderID = OrderDetails.OrderID
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> GROUP BY o.OrderID, c.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 |
+---------+------------+-----------+----------+-----------------+------------+---------+--------------+
|     108 |          8 | Om        | Kumar    | om123@gmail.com | 1111111136 | Bihar   |   1760008.72 |
+---------+------------+-----------+----------+-----------------+------------+---------+--------------+
1 row in set (0.01 sec)
```

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

```
mysql> SELECT
    ->     Products.ProductID,
    ->     Products.ProductName,
    ->     COUNT(OrderDetails.OrderID) AS NumberOfOrders
    -> FROM Products JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID
    -> WHERE  Products.Description = 'electronic gadget'
    -> GROUP BY Products.ProductID, Products.ProductName;
+-----------+-------------+----------------+
| ProductID | ProductName | NumberOfOrders |
+-----------+-------------+----------------+
|        11 | Laptop      |              1 |
|        12 | Mouse       |              1 |
|        13 | Keyboard    |              1 |
|        14 | Printer     |              1 |
|        15 | Phone       |              1 |
|        17 | Headphone   |              1 |
|        18 | Camera      |              1 |
+-----------+-------------+----------------+
7 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 FindCutsomers(
    ->
    -> IN ProductNameParam VARCHAR(255))
    ->
    -> BEGIN
    ->
    -> SELECT DISTINCT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     Customers.Email,
    ->     Customers.Phone,
    ->     Customers.Address
    -> FROM
    ->     Customers
    -> JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN
    ->     OrderDetails ON Orders.OrderID = OrderDetails.OrderID
    -> JOIN
    ->     Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE
    ->     Products.ProductName = ProductNameParam;
    ->  END ##
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql>  DELIMITER ;
mysql> call FindCustomers('Laptop');
ERROR 1305 (42000): PROCEDURE techshop.FindCustomers does not exist
mysql> DELIMITER ;
mysql> call FindCustomers('Laptop');
ERROR 1305 (42000): PROCEDURE techshop.FindCustomers does not exist
mysql> call FindCutsomers('Laptop');
+------------+-----------+----------+----------------+------------+---------+
| CustomerID | FirstName | LastName | Email          | Phone      | Address |
+------------+-----------+----------+----------------+------------+---------+
|          1 | Aman      | Kumar    | aman@gmail.com | 1234567897 | Motihar |
+------------+-----------+----------+----------------+------------+---------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.

```
mysql> DELIMITER ##
mysql> CREATE PROCEDURE CalculateTotalRevenue(
    ->
    -> IN StartDateParam date,
    -> IN EndDateParam date)
    ->
    -> BEGIN
    -> SELECT
    ->     SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue
    -> FROM
    ->     Orders
    -> JOIN
    ->     OrderDetails ON Orders.OrderID = OrderDetails.OrderID
    -> JOIN
    ->     Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE
    ->     Orders.OrderDate BETWEEN StartDateParam AND EndDateParam;
    ->  END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call CalculateTotalRevenue('2024-01-05', '2024-01-10');
+--------------+
| TotalRevenue |
+--------------+
|   1904415.21 |
+--------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 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 DISTINCT CustomerID FROM Orders);
+------------+-----------+--------------+------------------+------------+-------------+
| CustomerID | FirstName | LastName     | Email            | Phone      | Address     |
+------------+-----------+--------------+------------------+------------+-------------+
|          9 | Roshan    | Singh        | roshan@gmail.com | 1111111137 | Gandhinagar |
|         10 | Arava     | Krishnavenni | arava@gmail.com  | 1111111139 | Chennai     |
|         11 | Jasmine   | Roy          | jas123@gmail.com | 2258746984 | Mumbai      |
+------------+-----------+--------------+------------------+------------+-------------+
3 rows in set (0.00 sec)

mysql>
```

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

MySQL 8.0 Command Line Client

```
mysql> SELECT COUNT(*) AS TotalProducts
    -> FROM Products;
+---------------+
| TotalProducts |
+---------------+
|            11 |
+---------------+
1 row in set (0.03 sec)

mysql>
```

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

```
mysql> SELECT SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue
    -> FROM OrderDetails
    -> JOIN Products ON OrderDetails.ProductID = Products.ProductID
    -> JOIN Orders ON OrderDetails.OrderID = Orders.OrderID;
+--------------+
| TotalRevenue |
+--------------+
|   2493359.13 |
+--------------+
1 row in set (0.00 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> DELIMITER ##
mysql> CREATE PROCEDURE AvgQuantity(
    ->
    -> IN CategoryNameParam VARCHAR(255))
    ->
    -> BEGIN
    ->
    -> SELECT
    ->     AVG(OrderDetails.Quantity) AS AverageQuantityOrdered
    -> FROM
    ->     OrderDetails
    -> JOIN
    ->     Products ON OrderDetails.ProductID = Products.ProductID
    -> WHERE
    ->     Products.Description = CategoryNameParam;
    ->  END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call AvgQuantity('electronic gadget');
+------------------------+
| AverageQuantityOrdered |
+------------------------+
|      4.285714285714286 |
+------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> DELIMITER ##
mysql> CREATE PROCEDURE CalculateCustomerRevenue(
    ->
    -> IN CustomerIDParam int)
    ->
    -> BEGIN
    ->
    -> SELECT
    ->     SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue
    -> FROM
    ->     OrderDetails
    -> JOIN
    ->     Products ON OrderDetails.ProductID = Products.ProductID
    -> JOIN
    ->     Orders ON OrderDetails.OrderID = Orders.OrderID
    -> WHERE
    ->     Orders.CustomerID = CustomerIDParam;
    ->  END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> call CalculateCustomerRevenue(4);
+--------------+
| TotalRevenue |
+--------------+
|     35201.48 |
+--------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> SELECT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     COUNT(Orders.OrderID) AS NumberOfOrders
    -> FROM
    ->     Customers
    -> JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> ORDER BY
    ->     NumberOfOrders DESC
    -> LIMIT 1;
+------------+-----------+----------+----------------+
| CustomerID | FirstName | LastName | NumberOfOrders |
+------------+-----------+----------+----------------+
|          5 | Swati     | Singh    |              2 |
+------------+-----------+----------+----------------+
1 row in set (0.01 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> SELECT
    ->     Products.Description, SUM(OrderDetails.Quantity) AS TotalQuantityOrdered
    -> FROM
    ->     Products
    -> JOIN
    ->     OrderDetails ON Products.ProductID = OrderDetails.ProductID
    -> GROUP BY
    ->     Products.Description
    -> ORDER BY
    ->     TotalQuantityOrdered DESC
    -> LIMIT 1;
+------------------+----------------------+
| Description      | TotalQuantityOrdered |
+------------------+----------------------+
| electronic gadget |                   30 |
+------------------+----------------------+
1 row in set (0.00 sec)

mysql>
```

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 8.0 Command Line Client

```
mysql> SELECT
    ->     Customers.CustomerID,
    ->     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.Description = 'electronic gadget'
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> ORDER BY
    ->     TotalSpending DESC
    -> LIMIT 1;
+------------+-----------+----------+---------------+
| CustomerID | FirstName | LastName | TotalSpending |
+------------+-----------+----------+---------------+
|          8 | Om        | Kumar    |    1760008.72 |
+------------+-----------+----------+---------------+
1 row in set (0.00 sec)

mysql>
```

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

MySQL 8.0 Command Line Client

```
mysql> SELECT
    ->     Customers.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     COUNT(Orders.OrderID) AS NumberOfOrders,
    ->     SUM(OrderDetails.Quantity * Products.Price) AS TotalRevenue,
    ->     SUM(OrderDetails.Quantity * Products.Price) / COUNT(Orders.OrderID) AS AverageOrderValue
    -> FROM
    ->     Customers
    -> LEFT JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> LEFT JOIN
    ->     OrderDetails ON Orders.OrderID = OrderDetails.OrderID
    -> LEFT JOIN
    ->     Products ON OrderDetails.ProductID = Products.ProductID
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName;
+------------+-----------+--------------+----------------+--------------------+--------------------+
| CustomerID | FirstName | LastName     | NumberOfOrders | TotalRevenue       | AverageOrderValue  |
+------------+-----------+--------------+----------------+--------------------+--------------------+
|          1 | Aman      | Kumar        |              1 |          550000.15 |          550000.15 |
|          2 | Ansu      | Singh        |              1 |            1100.52 |            1100.52 |
|          3 | Amit      | Sah          |              1 |            2641.77 |            2641.77 |
|          4 | Asmita    | Si           |              1 |           35201.48 |           35201.48 |
|          5 | Swati     | Singh        |              2 |  99004.59999999999 | 49502.299999999996 |
|          6 | Prateeti  | Maji         |              1 | 30000.840000000004 | 30000.840000000004 |
|          7 | Anjali    | Sneha        |              1 | 15401.050000000001 | 15401.050000000001 |
|          8 | Om        | Kumar        |              1 |         1760008.72 |         1760008.72 |
|          9 | Roshan    | Singh        |              0 |               NULL |               NULL |
|         10 | Arava     | Krishnavenni |              0 |               NULL |               NULL |
|         11 | Jasmine   | Roy          |              0 |               NULL |               NULL |
+------------+-----------+--------------+----------------+--------------------+--------------------+
11 rows in set (0.01 sec)

mysql>
```

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.CustomerID,
    ->     Customers.FirstName,
    ->     Customers.LastName,
    ->     COUNT(Orders.OrderID) AS OrderCount
    -> FROM
    ->     Customers
    -> LEFT JOIN
    ->     Orders ON Customers.CustomerID = Orders.CustomerID
    -> GROUP BY
    ->     Customers.CustomerID, Customers.FirstName, Customers.LastName;
+------------+-----------+--------------+------------+
| CustomerID | FirstName | LastName     | OrderCount |
+------------+-----------+--------------+------------+
|          1 | Aman      | Kumar        |          1 |
|          2 | Ansu      | Singh        |          1 |
|          3 | Amit      | Sah          |          1 |
|          4 | Asmita    | Si           |          1 |
|          5 | Swati     | Singh        |          2 |
|          6 | Prateeti  | Maji         |          1 |
|          7 | Anjali    | Sneha        |          1 |
|          8 | Om        | Kumar        |          1 |
|          9 | Roshan    | Singh        |          0 |
|         10 | Arava     | Krishnavenni |          0 |
|         11 | Jasmine   | Roy          |          0 |
+------------+-----------+--------------+------------+
11 rows in set (0.00 sec)

mysql>
```