Task:1 Database Design

1.Create the database named "TechShop"

Query:

```
CREATE DATABASE TechShop;
USE TechShop;
```

```
mysql> CREATE DATABASE TechShop;
Query OK, 1 row affected (0.03 sec)

mysql> USE TechShop;
Database changed
```

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

Query:

```
CREATE TABLE Customers (
CustomerID INT,
FirstName VARCHAR(50),
LastName VARCHAR(50),
Email VARCHAR(100),
Phone VARCHAR(15),
Address VARCHAR(255));
```

```
mysql> CREATE TABLE Customers (
    ->     CustomerID INT,
    ->     FirstName VARCHAR(50),
    ->     LastName VARCHAR(50),
    ->     Email VARCHAR(100),
    ->     Phone VARCHAR(15),
    ->     Address VARCHAR(255));
Query OK, 0 rows affected (0.06 sec)

mysql> desc Customers;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| CustomerID | int          | YES  |     | NULL    |       |
| FirstName  | varchar(50)  | YES  |     | NULL    |       |
| LastName   | varchar(50)  | YES  |     | NULL    |       |
| Email      | varchar(100) | YES  |     | NULL    |       |
| Phone      | varchar(15)  | YES  |     | NULL    |       |
| Address    | varchar(255) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

CREATE TABLE Products (
ProductID INT,
ProductName VARCHAR(100),
Description TEXT,
Price DECIMAL(10, 2));

```
mysql> CREATE TABLE Products (
    ->     ProductID INT,
    ->     ProductName VARCHAR(100),
    ->     Description TEXT,
    ->     Price DECIMAL(10, 2));
Query OK, 0 rows affected (0.05 sec)

mysql> desc Products;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| ProductID   | int          | YES  |     | NULL    |       |
| ProductName | varchar(100) | YES  |     | NULL    |       |
| Description | text         | YES  |     | NULL    |       |
| Price       | decimal(10,2) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

CREATE TABLE Orders (
OrderID INT,
CustomerID INT,
OrderDate DATE,
TotalAmount DECIMAL(10, 2));

```
mysql> CREATE TABLE Orders (
    ->     OrderID INT,
    ->     CustomerID INT,
    ->     OrderDate DATE,
    ->     TotalAmount DECIMAL(10, 2));
Query OK, 0 rows affected (0.03 sec)

mysql> desc Orders;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| OrderID     | int          | YES  |     | NULL    |       |
| CustomerID  | int          | YES  |     | NULL    |       |
| OrderDate   | date         | YES  |     | NULL    |       |
| TotalAmount | decimal(10,2) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

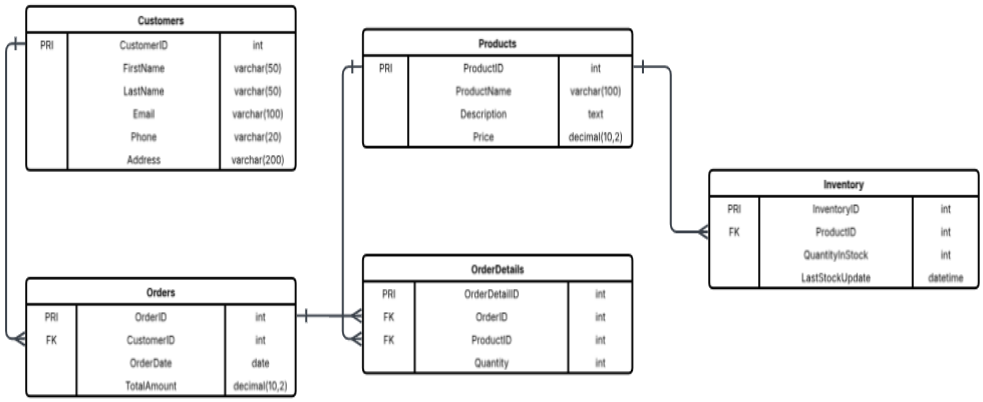
```
CREATE TABLE OrderDetails (
 OrderDetailID INT,
 OrderID INT,
 ProductID INT,
 Quantity INT);
```

```
mysql> CREATE TABLE OrderDetails (
    ->     OrderDetailID INT,
    ->     OrderID INT,
    ->     ProductID INT,
    ->     Quantity INT);
Query OK, 0 rows affected (0.03 sec)

mysql> desc OrderDetails;
+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| OrderDetailID | int  | YES  |     | NULL    |       |
| OrderID       | int  | YES  |     | NULL    |       |
| ProductID     | int  | YES  |     | NULL    |       |
| Quantity      | int  | YES  |     | NULL    |       |
+---------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

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

```
mysql> CREATE TABLE Inventory (
    ->     InventoryID INT,
    ->     ProductID INT,
    ->     QuantityInStock INT,
    ->     LastStockUpdate DATE);
Query OK, 0 rows affected (0.04 sec)

mysql> desc Inventory;
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| InventoryID     | int  | YES  |     | NULL    |       |
| ProductID       | int  | YES  |     | NULL    |       |
| QuantityInStock | int  | YES  |     | NULL    |       |
| LastStockUpdate | date | YES  |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

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



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

Query:

```
-- Altering tables to add primary keys
ALTER TABLE Customers ADD PRIMARY KEY (CustomerID);
ALTER TABLE Products ADD PRIMARY KEY (ProductID);
ALTER TABLE Orders ADD PRIMARY KEY (OrderID);
ALTER TABLE OrderDetails ADD PRIMARY KEY (OrderDetailID);
ALTER TABLE Inventory ADD PRIMARY KEY (InventoryID);
```

```
mysql> ALTER TABLE Customers ADD PRIMARY KEY (CustomerID);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Products ADD PRIMARY KEY (ProductID);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Orders ADD PRIMARY KEY (OrderID);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE OrderDetails ADD PRIMARY KEY (OrderDetailID);

Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE Inventory ADD PRIMARY KEY (InventoryID);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Query:

```
-- Adding foreign key constraints

ALTER TABLE Orders
ADD CONSTRAINT FK_Orders_Customers
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID);

ALTER TABLE OrderDetails
ADD CONSTRAINT FK_OrderDetails_Orders
FOREIGN KEY (OrderID) REFERENCES Orders(OrderID);

ALTER TABLE OrderDetails
ADD CONSTRAINT FK_OrderDetails_Products
FOREIGN KEY (ProductID) REFERENCES Products(ProductID);

ALTER TABLE Inventory
ADD CONSTRAINT FK_Inventory_Products
FOREIGN KEY (ProductID) REFERENCES Products(ProductID);
```

```
mysql> ALTER TABLE Orders
    -> ADD CONSTRAINT FK_Orders_Customers
    -> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
;
Query OK, 0 rows affected (0.12 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> ALTER TABLE OrderDetails
    -> ADD CONSTRAINT FK_OrderDetails_Orders
    -> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID);
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> ALTER TABLE OrderDetails
    -> ADD CONSTRAINT FK_OrderDetails_Products
    -> FOREIGN KEY (ProductID) REFERENCES Products(ProductID);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> ALTER TABLE Inventory
    -> ADD CONSTRAINT FK_Inventory_Products
    -> FOREIGN KEY (ProductID) REFERENCES Products(ProductID);
Query OK, 0 rows affected (0.12 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

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

a. Customers

Query:

```
INSERT INTO Customers VALUES
(1, 'Alice', 'Johnson', 'alice@email.com', '1234567890', 'Chennai'),
(2, 'Bob', 'Smith', 'bob@email.com', '9876543210', 'Mumbai'),
(3, 'Carol', 'Lee', 'carol@email.com', '7654321098', 'Delhi'),
(4, 'David', 'Kumar', 'david@email.com', '9090909090', 'Hyderabad'),
(5, 'Eva', 'Rao', 'eva@email.com', '8080808080', 'Bangalore'),
(6, 'Frank', 'Das', 'frank@email.com', '7070707070', 'Pune'),
(7, 'Grace', 'Menon', 'grace@email.com', '6060606060', 'Kolkata'),
(8, 'Henry', 'Singh', 'henry@email.com', '5050505050', 'Ahmedabad'),
(9, 'Ivy', 'Patel', 'ivy@email.com', '4040404040', 'Nagpur'),
(10, 'John', 'Fernandes', 'john@email.com', '3030303030', 'Trivandrum');
```

```
mysql> INSERT INTO Customers VALUES
    -> (1, 'Alice', 'Johnson', 'alice@email.com', '1234567890', 'Chennai'),
    -> (2, 'Bob', 'Smith', 'bob@email.com', '9876543210', 'Mumbai'),
    -> (3, 'Carol', 'Lee', 'carol@email.com', '7654321098', 'Delhi'),
    -> (4, 'David', 'Kumar', 'david@email.com', '9090909090', 'Hyderabad'),
    -> (5, 'Eva', 'Rao', 'eva@email.com', '8080808080', 'Bangalore'),
    -> (6, 'Frank', 'Das', 'frank@email.com', '7070707070', 'Pune'),
    -> (7, 'Grace', 'Menon', 'grace@email.com', '6060606060', 'Kolkata'),
    -> (8, 'Henry', 'Singh', 'henry@email.com', '5050505050', 'Ahmedabad'),
    -> (9, 'Ivy', 'Patel', 'ivy@email.com', '4040404040', 'Nagpur'),
    -> (10, 'John', 'Fernandes', 'john@email.com', '3030303030', 'Trivandrum');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Customers;
+------------+-----------+-----------+-----------------+------------+------------+
| CustomerID | FirstName | LastName  | Email           | Phone      | Address    |
+------------+-----------+-----------+-----------------+------------+------------+
|          1 | Alice     | Johnson   | alice@email.com | 1234567890 | Chennai    |
|          2 | Bob       | Smith     | bob@email.com   | 9876543210 | Mumbai     |
|          3 | Carol     | Lee       | carol@email.com | 7654321098 | Delhi      |
|          4 | David     | Kumar     | david@email.com | 9090909090 | Hyderabad  |
|          5 | Eva       | Rao       | eva@email.com   | 8080808080 | Bangalore  |
|          6 | Frank     | Das       | frank@email.com | 7070707070 | Pune       |
|          7 | Grace     | Menon     | grace@email.com | 6060606060 | Kolkata    |
|          8 | Henry     | Singh     | henry@email.com | 5050505050 | Ahmedabad  |
|          9 | Ivy       | Patel     | ivy@email.com   | 4040404040 | Nagpur     |
|         10 | John      | Fernandes | john@email.com  | 3030303030 | Trivandrum |
+------------+-----------+-----------+-----------------+------------+------------+
10 rows in set (0.00 sec)
```

b. Products

```
INSERT INTO Products VALUES
(101, 'Laptop', '14-inch, 8GB RAM, 512GB SSD', 55000.00),
(102, 'Smartphone', '6.5-inch, 128GB Storage', 25000.00),
(103, 'Wireless Mouse', 'Ergonomic, Bluetooth', 1200.00),
(104, 'Keyboard', 'Mechanical, RGB backlit', 2200.00),
(105, 'Monitor', '24-inch Full HD', 11000.00),
(106, 'External Hard Drive', '1TB USB 3.0', 4500.00),
(107, 'Tablet', '10-inch, 64GB Storage', 18000.00),
(108, 'Webcam', '1080p Full HD', 2300.00),
(109, 'Printer', 'Ink Tank, Color Printer', 9000.00),
(110, 'Headphones', 'Noise Cancelling, Over-ear', 3200.00);
```

```
mysql> INSERT INTO Products VALUES
    -> (101, 'Laptop', '14-inch, 8GB RAM, 512GB SSD', 55000.00),
    -> (102, 'Smartphone', '6.5-inch, 128GB Storage', 25000.00),
    -> (103, 'Wireless Mouse', 'Ergonomic, Bluetooth', 1200.00),
    -> (104, 'Keyboard', 'Mechanical, RGB backlit', 2200.00),
    -> (105, 'Monitor', '24-inch Full HD', 11000.00),
    -> (106, 'External Hard Drive', '1TB USB 3.0', 4500.00),
    -> (107, 'Tablet', '10-inch, 64GB Storage', 18000.00),
    -> (108, 'Webcam', '1080p Full HD', 2300.00),
    -> (109, 'Printer', 'Ink Tank, Color Printer', 9000.00),
    -> (110, 'Headphones', 'Noise Cancelling, Over-ear', 3200.00);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Products;
+-----------+--------------------+-----------------------------+----------+
| ProductID | ProductName        | Description                 | Price    |
+-----------+--------------------+-----------------------------+----------+
|       101 | Laptop             | 14-inch, 8GB RAM, 512GB SSD | 55000.00 |
|       102 | Smartphone         | 6.5-inch, 128GB Storage     | 25000.00 |
|       103 | Wireless Mouse     | Ergonomic, Bluetooth        |  1200.00 |
|       104 | Keyboard           | Mechanical, RGB backlit     |  2200.00 |
|       105 | Monitor            | 24-inch Full HD             | 11000.00 |
|       106 | External Hard Drive | 1TB USB 3.0                |  4500.00 |
|       107 | Tablet             | 10-inch, 64GB Storage       | 18000.00 |
|       108 | Webcam             | 1080p Full HD               |  2300.00 |
|       109 | Printer            | Ink Tank, Color Printer     |  9000.00 |
|       110 | Headphones         | Noise Cancelling, Over-ear  |  3200.00 |
+-----------+--------------------+-----------------------------+----------+
10 rows in set (0.02 sec)
```

c. Orders

```
INSERT INTO Orders VALUES
(201, 1, '2024-06-01', 58000.00),
(202, 2, '2024-06-02', 26000.00),
(203, 3, '2024-06-03', 13400.00),
(204, 4, '2024-06-04', 9000.00),
(205, 5, '2024-06-05', 11000.00),
(206, 6, '2024-06-06', 18000.00),
(207, 7, '2024-06-07', 5500.00),
(208, 8, '2024-06-08', 7700.00),
(209, 9, '2024-06-09', 12000.00),
(210, 10, '2024-06-10', 3500.00);
```

```
mysql> INSERT INTO Orders VALUES
    -> (201, 1, '2024-06-01', 58000.00),
    -> (202, 2, '2024-06-02', 26000.00),
    -> (203, 3, '2024-06-03', 13400.00),
    -> (204, 4, '2024-06-04', 9000.00),
    -> (205, 5, '2024-06-05', 11000.00),
    -> (206, 6, '2024-06-06', 18000.00),
    -> (207, 7, '2024-06-07', 5500.00),
    -> (208, 8, '2024-06-08', 7700.00),
    -> (209, 9, '2024-06-09', 12000.00),
    -> (210, 10, '2024-06-10', 3500.00);
Query OK, 10 rows affected (0.03 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     201 |          1 | 2024-06-01 |    58000.00 |
|     202 |          2 | 2024-06-02 |    26000.00 |
|     203 |          3 | 2024-06-03 |    13400.00 |
|     204 |          4 | 2024-06-04 |     9000.00 |
|     205 |          5 | 2024-06-05 |    11000.00 |
|     206 |          6 | 2024-06-06 |    18000.00 |
|     207 |          7 | 2024-06-07 |     5500.00 |
|     208 |          8 | 2024-06-08 |     7700.00 |
|     209 |          9 | 2024-06-09 |    12000.00 |
|     210 |         10 | 2024-06-10 |     3500.00 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

d. OrderDetails

```
INSERT INTO OrderDetails VALUES
(301, 201, 101, 1), -- Laptop
(302, 202, 102, 1), -- Smartphone
(303, 203, 103, 2), -- Wireless Mouse x2
(304, 203, 104, 1), -- Keyboard
(305, 204, 109, 1), -- Printer
(306, 205, 105, 1), -- Monitor
(307, 206, 107, 1), -- Tablet
(308, 207, 106, 1), -- External HDD
(309, 208, 104, 1), -- Keyboard
(310, 209, 108, 2); -- Webcam x2
```

```
mysql> INSERT INTO OrderDetails VALUES
    -> (301, 201, 101, 1), -- Laptop
    -> (302, 202, 102, 1), -- Smartphone
    -> (303, 203, 103, 2), -- Wireless Mouse x2
    -> (304, 203, 104, 1), -- Keyboard
    -> (305, 204, 109, 1), -- Printer
    -> (306, 205, 105, 1), -- Monitor
    -> (307, 206, 107, 1), -- Tablet
    -> (308, 207, 106, 1), -- External HDD
    -> (309, 208, 104, 1), -- Keyboard
    -> (310, 209, 108, 2); -- Webcam x2
Query OK, 10 rows affected (0.03 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from OrderDetails;;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|           301 |     201 |       101 |        1 |
|           302 |     202 |       102 |        1 |
|           303 |     203 |       103 |        2 |
|           304 |     203 |       104 |        1 |
|           305 |     204 |       109 |        1 |
|           306 |     205 |       105 |        1 |
|           307 |     206 |       107 |        1 |
|           308 |     207 |       106 |        1 |
|           309 |     208 |       104 |        1 |
|           310 |     209 |       108 |        2 |
+---------------+---------+-----------+----------+
10 rows in set (0.00 sec)
```

e. Inventory

```
INSERT INTO Inventory VALUES
(401, 101, 15, '2024-06-01'),
(402, 102, 25, '2024-06-01'),
(403, 103, 50, '2024-06-01'),
(404, 104, 40, '2024-06-01'),
(405, 105, 20, '2024-06-01'),
(406, 106, 30, '2024-06-01'),
(407, 107, 18, '2024-06-01'),
(408, 108, 35, '2024-06-01'),
(409, 109, 10, '2024-06-01'),
(410, 110, 28, '2024-06-01');
```

```
mysql> INSERT INTO Inventory VALUES
    -> (401, 101, 15, '2024-06-01'),
    -> (402, 102, 25, '2024-06-01'),
    -> (403, 103, 50, '2024-06-01'),
    -> (404, 104, 40, '2024-06-01'),
    -> (405, 105, 20, '2024-06-01'),
    -> (406, 106, 30, '2024-06-01'),
    -> (407, 107, 18, '2024-06-01'),
    -> (408, 108, 35, '2024-06-01'),
    -> (409, 109, 10, '2024-06-01'),
    -> (410, 110, 28, '2024-06-01');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from Inventory;
+-------------+-----------+-----------------+-----------------+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-------------+-----------+-----------------+-----------------+
|         401 |       101 |              15 | 2024-06-01      |
|         402 |       102 |              25 | 2024-06-01      |
|         403 |       103 |              50 | 2024-06-01      |
|         404 |       104 |              40 | 2024-06-01      |
|         405 |       105 |              20 | 2024-06-01      |
|         406 |       106 |              30 | 2024-06-01      |
|         407 |       107 |              18 | 2024-06-01      |
|         408 |       108 |              35 | 2024-06-01      |
|         409 |       109 |              10 | 2024-06-01      |
|         410 |       110 |              28 | 2024-06-01      |
+-------------+-----------+-----------------+-----------------+
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.

   Query:

   SELECT FirstName, LastName, Email
   FROM Customers;

```
mysql> SELECT FirstName, LastName, Email
    -> FROM Customers;
+-----------+-----------+-----------------+
| FirstName | LastName  | Email           |
+-----------+-----------+-----------------+
| Alice     | Johnson   | alice@email.com |
| Bob       | Smith     | bob@email.com   |
| Carol     | Lee       | carol@email.com |
| David     | Kumar     | david@email.com |
| Eva       | Rao       | eva@email.com   |
| Frank     | Das       | frank@email.com |
| Grace     | Menon     | grace@email.com |
| Henry     | Singh     | henry@email.com |
| Ivy       | Patel     | ivy@email.com   |
| John      | Fernandes | john@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.

   Query:
   SELECT Orders.OrderID, Orders.OrderDate, Customers.FirstName, Customers.LastName
   FROM Orders
   JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

```
mysql> SELECT Orders.OrderID, Orders.OrderDate, Customers.FirstName, Customers.LastName
    -> FROM Orders
    -> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
+---------+------------+-----------+-----------+
| OrderID | OrderDate  | FirstName | LastName  |
+---------+------------+-----------+-----------+
|     201 | 2024-06-01 | Alice     | Johnson   |
|     202 | 2024-06-02 | Bob       | Smith     |
|     203 | 2024-06-03 | Carol     | Lee       |
|     204 | 2024-06-04 | David     | Kumar     |
|     205 | 2024-06-05 | Eva       | Rao       |
|     206 | 2024-06-06 | Frank     | Das       |
|     207 | 2024-06-07 | Grace     | Menon     |
|     208 | 2024-06-08 | Henry     | Singh     |
|     209 | 2024-06-09 | Ivy       | Patel     |
|     210 | 2024-06-10 | John      | Fernandes |
+---------+------------+-----------+-----------+
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.

   Query:
   INSERT into Customers VALUES
   (11,'Ravi', 'Sharma', 'ravi@gmail.com', '9999999999', 'Chennai');

```
mysql> INSERT into Customers VALUES(11,'Ravi', 'Sharma', 'ravi@gmail.com', '999999
9999', 'Chennai');
Query OK, 1 row affected (0.02 sec)

mysql> select * from customers;
+------------+-----------+-----------+-----------------+------------+------------+
| CustomerID | FirstName | LastName  | Email           | Phone      | Address    |
+------------+-----------+-----------+-----------------+------------+------------+
|          1 | Alice     | Johnson   | alice@email.com | 1234567890 | Chennai    |
|          2 | Bob       | Smith     | bob@email.com   | 9876543210 | Mumbai     |
|          3 | Carol     | Lee       | carol@email.com | 7654321098 | Delhi      |
|          4 | David     | Kumar     | david@email.com | 9090909090 | Hyderabad  |
|          5 | Eva       | Rao       | eva@email.com   | 8080808080 | Bangalore  |
|          6 | Frank     | Das       | frank@email.com | 7070707070 | Pune       |
|          7 | Grace     | Menon     | grace@email.com | 6060606060 | Kolkata    |
|          8 | Henry     | Singh     | henry@email.com | 5050505050 | Ahmedabad  |
|          9 | Ivy       | Patel     | ivy@email.com   | 4040404040 | Nagpur     |
|         10 | John      | Fernandes | john@email.com  | 3030303030 | Trivandrum |
|         11 | Ravi      | Sharma    | ravi@gmail.com  | 9999999999 | Chennai    |
+------------+-----------+-----------+-----------------+------------+------------+
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%.

   Query:
   update products set price = price * 1.10;

```
mysql> update products set price=price*1.10;
Query OK, 10 rows affected (0.03 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from products;
+-----------+---------------------+-----------------------------+----------+
| ProductID | ProductName         | Description                 | Price    |
+-----------+---------------------+-----------------------------+----------+
|       101 | Laptop              | 14-inch, 8GB RAM, 512GB SSD | 66550.00 |
|       102 | Smartphone          | 6.5-inch, 128GB Storage     | 30250.00 |
|       103 | Wireless Mouse      | Ergonomic, Bluetooth        |  1452.00 |
|       104 | Keyboard            | Mechanical, RGB backlit     |  2662.00 |
|       105 | Monitor             | 24-inch Full HD             | 13310.00 |
|       106 | External Hard Drive | 1TB USB 3.0                 |  4950.00 |
|       107 | Tablet              | 10-inch, 64GB Storage       | 21780.00 |
|       108 | Webcam              | 1080p Full HD               |  2530.00 |
|       109 | Printer             | Ink Tank, Color Printer     | 10890.00 |
|       110 | Headphones          | Noise Cancelling, Over-ear  |  3872.00 |
+-----------+---------------------+-----------------------------+----------+
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.

Query:

```
DELETE FROM OrderDetails WHERE OrderID=210;
DELETE FROM Orders WHERE OrderID=210;
```

```
mysql> DELETE FROM OrderDetails WHERE OrderID=210;
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM Orders WHERE OrderID=210;
Query OK, 1 row affected (0.01 sec)

mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     201 |          1 | 2024-06-01 |    58000.00 |
|     202 |          2 | 2024-06-02 |    26000.00 |
|     203 |          3 | 2024-06-03 |    13400.00 |
|     204 |          4 | 2024-06-04 |     9000.00 |
|     205 |          5 | 2024-06-05 |    11000.00 |
|     206 |          6 | 2024-06-06 |    18000.00 |
|     207 |          7 | 2024-06-07 |     5500.00 |
|     208 |          8 | 2024-06-08 |     7700.00 |
|     209 |          9 | 2024-06-09 |    12000.00 |
+---------+------------+------------+-------------+
9 rows in set (0.00 sec)
```

```
mysql> select * from OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|           301 |     201 |       101 |        1 |
|           302 |     202 |       102 |        1 |
|           303 |     203 |       103 |        2 |
|           304 |     203 |       104 |        1 |
|           305 |     204 |       109 |        1 |
|           306 |     205 |       105 |        1 |
|           307 |     206 |       107 |        1 |
|           308 |     207 |       106 |        1 |
|           309 |     208 |       104 |        1 |
|           310 |     209 |       108 |        2 |
+---------------+---------+-----------+----------+
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.

   Query:
   INSERT into Orders VALUES
   (211, 11, '2024-06-11', 14500.00);

```
mysql> INSERT into Orders VALUES (211, 11, '2024-06-11', 14500.00);
Query OK, 1 row affected (0.03 sec)

mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     201 |          1 | 2024-06-01 |    58000.00 |
|     202 |          2 | 2024-06-02 |    26000.00 |
|     203 |          3 | 2024-06-03 |    13400.00 |
|     204 |          4 | 2024-06-04 |     9000.00 |
|     205 |          5 | 2024-06-05 |    11000.00 |
|     206 |          6 | 2024-06-06 |    18000.00 |
|     207 |          7 | 2024-06-07 |     5500.00 |
|     208 |          8 | 2024-06-08 |     7700.00 |
|     209 |          9 | 2024-06-09 |    12000.00 |
|     211 |         11 | 2024-06-11 |    14500.00 |
+---------+------------+------------+-------------+
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.

   Query:
   UPDATE Customers
   SET Email='@04alice@gmail.com', Address='Vellore'
   WHERE CustomerID=1;

```
mysql> UPDATE Customers
    -> SET Email='@04alice@gmail.com', Address='Vellore'
    -> WHERE CustomerID=1;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Customers;
+------------+-----------+-----------+--------------------+------------+------------+
| CustomerID | FirstName | LastName  | Email              | Phone      | Address    |
+------------+-----------+-----------+--------------------+------------+------------+
|          1 | Alice     | Johnson   | @04alice@gmail.com | 1234567890 | Vellore    |
|          2 | Bob       | Smith     | bob@email.com      | 9876543210 | Mumbai     |
|          3 | Carol     | Lee       | carol@email.com    | 7654321098 | Delhi      |
|          4 | David     | Kumar     | david@email.com    | 9090909090 | Hyderabad  |
|          5 | Eva       | Rao       | eva@email.com      | 8080808080 | Bangalore  |
|          6 | Frank     | Das       | frank@email.com    | 7070707070 | Pune       |
|          7 | Grace     | Menon     | grace@email.com    | 6060606060 | Kolkata    |
|          8 | Henry     | Singh     | henry@email.com    | 5050505050 | Ahmedabad  |
|          9 | Ivy       | Patel     | ivy@email.com      | 4040404040 | Nagpur     |
|         10 | John      | Fernandes | john@email.com     | 3030303030 | Trivandrum |
|         11 | Ravi      | Sharma    | ravi@gmail.com     | 9999999999 | Chennai    |
+------------+-----------+-----------+--------------------+------------+------------+
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.

Query:

```
UPDATE Orders
SET TotalAmount = (
SELECT SUM(OD.Quantity * P.Price)
FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
WHERE OD.OrderID = Orders.OrderID );
```

```
mysql> UPDATE Orders
    -> SET TotalAmount = (
    ->     SELECT SUM(OD.Quantity * P.Price)
    ->     FROM OrderDetails OD
    ->     JOIN Products P ON OD.ProductID = P.ProductID
    ->     WHERE OD.OrderID = Orders.OrderID
    -> );
Query OK, 1 row affected (0.01 sec)
Rows matched: 10  Changed: 1  Warnings: 0

mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     201 |          1 | 2024-06-01 |    66550.00 |
|     202 |          2 | 2024-06-02 |    30250.00 |
|     203 |          3 | 2024-06-03 |     5566.00 |
|     204 |          4 | 2024-06-04 |    10890.00 |
|     205 |          5 | 2024-06-05 |    13310.00 |
|     206 |          6 | 2024-06-06 |    21780.00 |
|     207 |          7 | 2024-06-07 |     4950.00 |
|     208 |          8 | 2024-06-08 |     2662.00 |
|     209 |          9 | 2024-06-09 |     5060.00 |
|     211 |         11 | 2024-06-11 |        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.

Query:

```
DELETE FROM OrderDetails
WHERE OrderID IN (
SELECT OrderID FROM Orders WHERE CustomerID = 3);

DELETE FROM Orders
WHERE CustomerID = 3;
```

```
mysql> DELETE FROM OrderDetails
    -> WHERE OrderID IN (
    ->     SELECT OrderID FROM Orders WHERE CustomerID = 3
    -> );
Query OK, 2 rows affected (0.01 sec)

mysql>
mysql> DELETE FROM Orders
    -> WHERE CustomerID = 3;
Query OK, 1 row affected (0.00 sec)

mysql> Select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|     201 |          1 | 2024-06-01 |    66550.00 |
|     202 |          2 | 2024-06-02 |    30250.00 |
|     204 |          4 | 2024-06-04 |    10890.00 |
|     205 |          5 | 2024-06-05 |    13310.00 |
|     206 |          6 | 2024-06-06 |    21780.00 |
|     207 |          7 | 2024-06-07 |     4950.00 |
|     208 |          8 | 2024-06-08 |     2662.00 |
|     209 |          9 | 2024-06-09 |     5060.00 |
|     211 |         11 | 2024-06-11 |        NULL |
+---------+------------+------------+-------------+
9 rows in set (0.00 sec)

mysql> Select * from OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|           301 |     201 |       101 |        1 |
|           302 |     202 |       102 |        1 |
|           305 |     204 |       109 |        1 |
|           306 |     205 |       105 |        1 |
|           307 |     206 |       107 |        1 |
|           308 |     207 |       106 |        1 |
|           309 |     208 |       104 |        1 |
|           310 |     209 |       108 |        2 |
+---------------+---------+-----------+----------+
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.

Query:

```
INSERT INTO Products VALUES
(111, 'Smartwatch', 'Bluetooth, Waterproof, Touchscreen', 4500.00);
```

```
mysql> INSERT INTO Products VALUES
    -> (111, 'Smartwatch', 'Bluetooth, Waterproof, Touchscreen', 4500.00);
Query OK, 1 row affected (0.03 sec)

mysql> Select * from Products;
+-----------+---------------------+-------------------------------------+----------+
| ProductID | ProductName         | Description                         | Price    |
+-----------+---------------------+-------------------------------------+----------+
|       101 | Laptop              | 14-inch, 8GB RAM, 512GB SSD         | 66550.00 |
|       102 | Smartphone          | 6.5-inch, 128GB Storage             | 30250.00 |
|       103 | Wireless Mouse      | Ergonomic, Bluetooth                |  1452.00 |
|       104 | Keyboard            | Mechanical, RGB backlit             |  2662.00 |
|       105 | Monitor             | 24-inch Full HD                     | 13310.00 |
|       106 | External Hard Drive | 1TB USB 3.0                         |  4950.00 |
|       107 | Tablet              | 10-inch, 64GB Storage               | 21780.00 |
|       108 | Webcam              | 1080p Full HD                       |  2530.00 |
|       109 | Printer             | Ink Tank, Color Printer             | 10890.00 |
|       110 | Headphones          | Noise Cancelling, Over-ear          |  3872.00 |
|       111 | Smartwatch          | Bluetooth, Waterproof, Touchscreen  |  4500.00 |
+-----------+---------------------+-------------------------------------+----------+
11 rows in set (0.00 sec)
```

11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.

Query:

```
ALTER TABLE Orders ADD Status VARCHAR(20);

UPDATE Orders
SET Status = 'Shipped'
WHERE OrderID = 201;
```

```
mysql> ALTER TABLE Orders
    -> ADD Status VARCHAR(20);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> UPDATE Orders
    -> SET Status = 'Shipped'
    -> WHERE OrderID = 201;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> Select * from Orders;
+---------+------------+------------+-------------+---------+
| OrderID | CustomerID | OrderDate  | TotalAmount | Status  |
+---------+------------+------------+-------------+---------+
|     201 |          1 | 2024-06-01 |    66550.00 | Shipped |
|     202 |          2 | 2024-06-02 |    30250.00 | NULL    |
|     204 |          4 | 2024-06-04 |    10890.00 | NULL    |
|     205 |          5 | 2024-06-05 |    13310.00 | NULL    |
|     206 |          6 | 2024-06-06 |    21780.00 | NULL    |
|     207 |          7 | 2024-06-07 |     4950.00 | NULL    |
|     208 |          8 | 2024-06-08 |     2662.00 | NULL    |
|     209 |          9 | 2024-06-09 |     5060.00 | NULL    |
|     211 |         11 | 2024-06-11 |        NULL | NULL    |
+---------+------------+------------+-------------+---------+
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.

Query:

```
ALTER TABLE Customers
ADD NumberOfOrders INT DEFAULT 0;

UPDATE Customers c
SET NumberOfOrders = (
 SELECT COUNT(*)
 FROM Orders o
 WHERE o.CustomerID = c.CustomerID );
```

```
mysql> ALTER TABLE Customers
    -> ADD NumberOfOrders INT DEFAULT 0;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> UPDATE Customers c
    -> SET NumberOfOrders = (
    ->     SELECT COUNT(*)
    ->     FROM Orders o
    ->     WHERE o.CustomerID = c.CustomerID
    -> );
Query OK, 9 rows affected (0.03 sec)
Rows matched: 11  Changed: 9  Warnings: 0

mysql> Select * from Customers;
+------------+-----------+-----------+--------------------+------------+------------+----------------+
| CustomerID | FirstName | LastName  | Email              | Phone      | Address    | NumberOfOrders |
+------------+-----------+-----------+--------------------+------------+------------+----------------+
|          1 | Alice     | Johnson   | @04alice@gmail.com | 1234567890 | Vellore    |              1 |
|          2 | Bob       | Smith     | bob@email.com      | 9876543210 | Mumbai     |              1 |
|          3 | Carol     | Lee       | carol@email.com    | 7654321098 | Delhi      |              0 |
|          4 | David     | Kumar     | david@email.com    | 9090909090 | Hyderabad  |              1 |
|          5 | Eva       | Rao       | eva@email.com      | 8080808080 | Bangalore  |              1 |
|          6 | Frank     | Das       | frank@email.com    | 7070707070 | Pune       |              1 |
|          7 | Grace     | Menon     | grace@email.com    | 6060606060 | Kolkata    |              1 |
|          8 | Henry     | Singh     | henry@email.com    | 5050505050 | Ahmedabad  |              1 |
|          9 | Ivy       | Patel     | ivy@email.com      | 4040404040 | Nagpur     |              1 |
|         10 | John      | Fernandes | john@email.com     | 3030303030 | Trivandrum |              0 |
|         11 | Ravi      | Sharma    | ravi@gmail.com     | 9999999999 | Chennai    |              1 |
+------------+-----------+-----------+--------------------+------------+------------+----------------+
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.

Query:

```
SELECT o.OrderID, o.OrderDate, c.FirstName, c.LastName
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID;
```

```
mysql> SELECT o.OrderID, o.OrderDate, c.FirstName, c.LastName
    -> FROM Orders o
    -> JOIN Customers c ON o.CustomerID = c.CustomerID;
+---------+------------+-----------+----------+
| OrderID | OrderDate  | FirstName | LastName |
+---------+------------+-----------+----------+
|     201 | 2024-06-01 | Alice     | Johnson  |
|     202 | 2024-06-02 | Bob       | Smith    |
|     204 | 2024-06-04 | David     | Kumar    |
|     205 | 2024-06-05 | Eva       | Rao      |
|     206 | 2024-06-06 | Frank     | Das      |
|     207 | 2024-06-07 | Grace     | Menon    |
|     208 | 2024-06-08 | Henry     | Singh    |
|     209 | 2024-06-09 | Ivy       | Patel    |
|     211 | 2024-06-11 | Ravi      | Sharma   |
+---------+------------+-----------+----------+
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.

Query:

```
SELECT p.ProductName, SUM(od.Quantity * p.Price) AS TotalRevenue
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductID, p.ProductName;
```

```
mysql> SELECT p.ProductName, SUM(od.Quantity * p.Price) AS TotalRevenue
    -> FROM OrderDetails od
    -> JOIN Products p ON od.ProductID = p.ProductID
    -> GROUP BY p.ProductID, p.ProductName;
+--------------------+--------------+
| ProductName        | TotalRevenue |
+--------------------+--------------+
| Laptop             |     66550.00 |
| Smartphone         |     30250.00 |
| Printer            |     10890.00 |
| Monitor            |     13310.00 |
| Tablet             |     21780.00 |
| External Hard Drive |      4950.00 |
| Keyboard           |      2662.00 |
| Webcam             |      5060.00 |
+--------------------+--------------+
8 rows in set (0.01 sec)
```

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

Query:

```
SELECT c.FirstName,  c.LastName, c.Email, c.Phone,
COUNT(o.OrderID) AS NumberOfPurchases
FROM  Customers c
JOIN  Orders o ON c.CustomerID = o.CustomerID
GROUP BY
c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone
HAVING
 COUNT(o.OrderID) >= 1;
```

```
mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount, Status)
    -> VALUES
    -> (10, 2, '2024-06-01', 299.99,'shipped'),(11, 2, '2024-06-10', 149.50,'shipped'),(12, 6, '2024-06-11', 450.00,'shipped');
Query OK, 3 rows affected (0.05 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT
    ->     c.FirstName,
    ->     c.LastName,
    ->     c.Email,
    ->     c.Phone,
    ->     COUNT(o.OrderID) AS NumberOfPurchases
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Orders o ON c.CustomerID = o.CustomerID
    -> GROUP BY
    ->     c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone
    -> HAVING
    ->     COUNT(o.OrderID) >= 1;
+-----------+----------+------------------+------------+------------------+
| FirstName | LastName | Email            | Phone      | NumberOfPurchases |
+-----------+----------+------------------+------------+------------------+
| Alice     | Johnson  | @04alice@gmail.com | 1234567890 |                1 |
| Bob       | Smith    | bob@email.com      | 9876543210 |                3 |
| David     | Kumar    | david@email.com    | 9090909090 |                1 |
| Eva       | Rao      | eva@email.com      | 8080808080 |                1 |
| Frank     | Das      | frank@email.com    | 7070707070 |                2 |
| Grace     | Menon    | grace@email.com    | 6060606060 |                1 |
| Henry     | Singh    | henry@email.com    | 5050505050 |                1 |
| Ivy       | Patel    | ivy@email.com      | 4040404040 |                1 |
| Ravi      | Sharma   | ravi@gmail.com     | 9999999999 |                1 |
+-----------+----------+------------------+------------+------------------+
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.

Query:

```
SELECT p.ProductName, SUM(od.Quantity) AS TotalOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductID, p.ProductName
ORDER BY TotalOrdered DESC
LIMIT 1;
```

```
mysql> SELECT p.ProductName, SUM(od.Quantity) AS TotalOrdered
    -> FROM OrderDetails od
    -> JOIN Products p ON od.ProductID = p.ProductID
    -> GROUP BY p.ProductID, p.ProductName
    -> ORDER BY TotalOrdered DESC
    -> LIMIT 1;
+-------------+--------------+
| ProductName | TotalOrdered |
+-------------+--------------+
| Webcam      |            2 |
+-------------+--------------+
1 row in set (0.00 sec)
```

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

Query:

```
ALTER TABLE Products
ADD Category VARCHAR(100);

UPDATE Products SET Category = 'Computers' WHERE ProductID = 101;
UPDATE Products SET Category = 'Mobile Devices' WHERE ProductID = 102;
UPDATE Products SET Category = 'Accessories' WHERE ProductID = 103;
UPDATE Products SET Category = 'Accessories' WHERE ProductID = 104;
UPDATE Products SET Category = 'Display' WHERE ProductID = 105;
UPDATE Products SET Category = 'Storage Devices' WHERE ProductID = 106;
UPDATE Products SET Category = 'Mobile Devices' WHERE ProductID = 107;
UPDATE Products SET Category = 'Accessories' WHERE ProductID = 108;
UPDATE Products SET Category = 'Printers' WHERE ProductID = 109;
UPDATE Products SET Category = 'Audio Devices' WHERE ProductID = 110;
UPDATE Products SET Category = 'Wearables' WHERE ProductID = 111;

SELECT ProductName, Category FROM Products;
```

```
mysql> ALTER TABLE Products
    -> ADD Category VARCHAR(100);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> UPDATE Products SET Category = 'Computers' WHERE ProductID = 101;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Mobile Devices' WHERE ProductID = 102;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Accessories' WHERE ProductID = 103;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Accessories' WHERE ProductID = 104;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Display' WHERE ProductID = 105;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Storage Devices' WHERE ProductID = 106;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Mobile Devices' WHERE ProductID = 107;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Accessories' WHERE ProductID = 108;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> UPDATE Products SET Category = 'Printers' WHERE ProductID = 109;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Audio Devices' WHERE ProductID = 110;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE Products SET Category = 'Wearables' WHERE ProductID = 111;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT ProductName, Category
    -> FROM Products;
+---------------------+-----------------+
| ProductName         | Category        |
+---------------------+-----------------+
| Laptop              | Computers       |
| Smartphone          | Mobile Devices  |
| Wireless Mouse      | Accessories     |
| Keyboard            | Accessories     |
| Monitor             | Display         |
| External Hard Drive | Storage Devices |
| Tablet              | Mobile Devices  |
| Webcam              | Accessories     |
| Printer             | Printers        |
| Headphones          | Audio Devices   |
| Smartwatch          | Wearables       |
+---------------------+-----------------+
11 rows in set (0.00 sec)
```

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

Query:

```
SELECT c.FirstName, c.LastName, AVG(o.TotalAmount) AS AvgOrderValue
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
GROUP BY c.CustomerID, c.FirstName, c.LastName;
```

```
mysql> SELECT c.FirstName, c.LastName, AVG(o.TotalAmount) AS AvgOrderValue
    -> FROM Orders o
    -> JOIN Customers c ON o.CustomerID = c.CustomerID
    -> GROUP BY c.CustomerID, c.FirstName, c.LastName;
+-----------+----------+---------------+
| FirstName | LastName | AvgOrderValue |
+-----------+----------+---------------+
| Alice     | Johnson  |  66550.000000 |
| Bob       | Smith    |  30250.000000 |
| David     | Kumar    |  10890.000000 |
| Eva       | Rao      |  13310.000000 |
| Frank     | Das      |  21780.000000 |
| Grace     | Menon    |   4950.000000 |
| Henry     | Singh    |   2662.000000 |
| Ivy       | Patel    |   5060.000000 |
| Ravi      | Sharma   |          NULL |
+-----------+----------+---------------+
9 rows in set (0.03 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.

Query:

```
SELECT o.OrderID, c.FirstName, c.LastName, o.TotalAmount
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
ORDER BY o.TotalAmount DESC
LIMIT 1;
```

```
mysql> SELECT o.OrderID, c.FirstName, c.LastName, o.TotalAmount
    -> FROM Orders o
    -> JOIN Customers c ON o.CustomerID = c.CustomerID
    -> ORDER BY o.TotalAmount DESC
    -> LIMIT 1;
+---------+-----------+----------+-------------+
| OrderID | FirstName | LastName | TotalAmount |
+---------+-----------+----------+-------------+
|     201 | Alice     | Johnson  |    66550.00 |
+---------+-----------+----------+-------------+
1 row in set (0.00 sec)
```

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

Query:

```
SELECT p.ProductName, COUNT(od.OrderDetailID) AS TimesOrdered
FROM OrderDetails od
 JOIN Products p ON od.ProductID = p.ProductID
 GROUP BY p.ProductID, p.ProductName;
```

```
mysql> SELECT p.ProductName, COUNT(od.OrderDetailID) AS TimesOrdered
    -> FROM OrderDetails od
    ->  JOIN Products p ON od.ProductID = p.ProductID
    ->  GROUP BY p.ProductID, p.ProductName;
+--------------------+--------------+
| ProductName        | TimesOrdered |
+--------------------+--------------+
| Laptop             |            1 |
| Smartphone         |            1 |
| Wireless Mouse     |            1 |
| Keyboard           |            1 |
| Monitor            |            2 |
| External Hard Drive |            1 |
| Tablet             |            1 |
| Webcam             |            1 |
| Printer            |            1 |
+--------------------+--------------+
9 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.

Query:

```
SELECT DISTINCT c.FirstName, c.LastName, c.Email, c.Phone
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.ProductName = 'laptop';
```

```
mysql> SELECT DISTINCT c.FirstName, c.LastName, c.Email, c.Phone
    -> FROM Customers c
    -> JOIN Orders o ON c.CustomerID = o.CustomerID
    -> JOIN OrderDetails od ON o.OrderID = od.OrderID
    -> JOIN Products p ON od.ProductID = p.ProductID
    -> WHERE p.ProductName = 'laptop';
+-----------+----------+-------------------+------------+
| FirstName | LastName | Email             | Phone      |
+-----------+----------+-------------------+------------+
| Alice     | Johnson  | @04alice@gmail.com | 1234567890 |
+-----------+----------+-------------------+------------+
1 row in set (0.00 sec)
```

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

Query:

```
SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders
WHERE OrderDate BETWEEN '2024-01-01' AND '2024-12-31';
```

```
mysql> SELECT SUM(TotalAmount) AS TotalRevenue
    -> FROM Orders
    -> WHERE OrderDate BETWEEN '2024-01-01' AND '2024-12-31';
+--------------+
| TotalRevenue |
+--------------+
|    155452.00 |
+--------------+
1 row in set (0.02 sec)
```

Task 4. Subquery and its type

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

Query:

```
SELECT FirstName, LastName
FROM Customers
WHERE CustomerID NOT IN (
SELECT DISTINCT CustomerID FROM Orders);
```

```
mysql> SELECT FirstName, LastName
    -> FROM Customers
    -> WHERE CustomerID NOT IN (
    ->     SELECT DISTINCT CustomerID FROM Orders
    -> );
+-----------+-----------+
| FirstName | LastName  |
+-----------+-----------+
| Carol     | Lee       |
| John      | Fernandes |
+-----------+-----------+
2 rows in set (0.02 sec)
```

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

Query:

```
SELECT COUNT(*) AS TotalProducts
FROM Products;
```

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

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

Query:

```
SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders
WHERE TotalAmount IS NOT NULL;
```

```
mysql> SELECT SUM(TotalAmount) AS TotalRevenue
    -> FROM Orders
    -> WHERE TotalAmount IS NOT NULL;
+--------------+
| TotalRevenue |
+--------------+
|    155452.00 |
+--------------+
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.

Query:

```
SELECT AVG(od.Quantity) AS AvgQuantityOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.Category = 'Mobile Devices';
```

```
mysql> SELECT AVG(od.Quantity) AS AvgQuantityOrdered
    -> FROM OrderDetails od
    -> JOIN Products p ON od.ProductID = p.ProductID
    -> WHERE p.Category = 'Mobile Devices';
+--------------------+
| AvgQuantityOrdered |
+--------------------+
|             1.0000 |
+--------------------+
1 row in set (0.02 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.

Query:

```
SELECT SUM(TotalAmount) AS CustomerRevenue
FROM Orders
WHERE CustomerID = 1;
```

```
mysql> SELECT SUM(TotalAmount) AS CustomerRevenue
    -> FROM Orders
    -> WHERE CustomerID = 1;
+-----------------+
| CustomerRevenue |
+-----------------+
|        66550.00 |
+-----------------+
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.

Query:

```
UPDATE Customers c
SET NumberOfOrders = (
SELECT COUNT(*)
FROM Orders o
WHERE o.CustomerID = c.CustomerID);

WITH OrderCounts AS (
  SELECT
    CustomerID,
    COUNT(*) AS TotalOrders
  FROM Orders
  GROUP BY CustomerID
)
SELECT c.CustomerID, c.FirstName, c.LastName, oc.TotalOrders
FROM Customers c
JOIN OrderCounts oc ON c.CustomerID = oc.CustomerID
WHERE oc.TotalOrders = (
  SELECT MAX(TotalOrders) FROM OrderCounts
);
```

```
mysql> UPDATE Customers c
    -> SET NumberOfOrders = (
    ->     SELECT COUNT(*)
    ->     FROM Orders o
    ->     WHERE o.CustomerID = c.CustomerID
    -> );
Query OK, 2 rows affected (0.05 sec)
Rows matched: 11  Changed: 2  Warnings: 0

mysql> WITH OrderCounts AS (
    ->     SELECT
    ->         CustomerID,
    ->         COUNT(*) AS TotalOrders
    ->     FROM Orders
    ->     GROUP BY CustomerID
    -> )
    -> SELECT c.CustomerID, c.FirstName, c.LastName, oc.TotalOrders
    -> FROM Customers c
    -> JOIN OrderCounts oc ON c.CustomerID = oc.CustomerID
    -> WHERE oc.TotalOrders = (
    ->     SELECT MAX(TotalOrders) FROM OrderCounts
    -> );
+------------+-----------+----------+-------------+
| CustomerID | FirstName | LastName | TotalOrders |
+------------+-----------+----------+-------------+
|          2 | Bob       | Smith    |           3 |
+------------+-----------+----------+-------------+
1 row in set (0.00 sec)
```

7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.

Query:

```
SELECT Category, SUM(od.Quantity) AS TotalOrdered
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY Category
ORDER BY TotalOrdered DESC
LIMIT 1;
```

```
mysql> SELECT Category, SUM(od.Quantity) AS TotalOrdered
    -> FROM OrderDetails od
    -> JOIN Products p ON od.ProductID = p.ProductID
    -> GROUP BY Category
    -> ORDER BY TotalOrdered DESC
    -> LIMIT 1;
+-------------+--------------+
| Category    | TotalOrdered |
+-------------+--------------+
| Accessories |            3 |
+-------------+--------------+
1 row in set (0.00 sec)
```

8. Write an SQL query to find the customer who has spent the most money (highest total revenue) on electronic gadgets. List their name and total spending.

Query:

```
SELECT c.FirstName, c.LastName, SUM(od.Quantity * p.Price) AS TotalSpent
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.Category IN ('Mobile Devices', 'Computers', 'Wearables', 'Storage Devices',
'Printers', 'Display', 'Audio Devices')
GROUP BY c.CustomerID
ORDER BY TotalSpent DESC
LIMIT 1;
```

```
mysql> SELECT c.FirstName, c.LastName, SUM(od.Quantity * p.Price) AS TotalSpent
    -> FROM Customers c
    -> JOIN Orders o ON c.CustomerID = o.CustomerID
    -> JOIN OrderDetails od ON o.OrderID = od.OrderID
    -> JOIN Products p ON od.ProductID = p.ProductID
    -> WHERE p.Category IN ('Mobile Devices', 'Computers', 'Wearables', 'Storage Devices', 'Printers', 'Display', 'Audio Devices')
    -> GROUP BY c.CustomerID
    -> ORDER BY TotalSpent DESC
    -> LIMIT 1;
+-----------+----------+------------+
| FirstName | LastName | TotalSpent |
+-----------+----------+------------+
| Alice     | Johnson  |   66550.00 |
+-----------+----------+------------+
1 row in set (0.00 sec)
```

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

Query:

```
SELECT ROUND(SUM(TotalAmount) / COUNT(OrderID), 2) AS AverageOrderValue
FROM Orders WHERE TotalAmount IS NOT NULL;
```

```
mysql> SELECT
    ->     ROUND(SUM(TotalAmount) / COUNT(OrderID), 2) AS AverageOrderValue
    -> FROM
    ->     Orders
    -> WHERE
    ->     TotalAmount IS NOT NULL;
+-------------------+
| AverageOrderValue |
+-------------------+
|          14213.77 |
+-------------------+
1 row in set (0.01 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.

Query:

```
SELECT c.FirstName, c.LastName, COUNT(o.OrderID) AS OrderCount
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID;
```

```
mysql> SELECT c.FirstName, c.LastName, COUNT(o.OrderID) AS OrderCount
    -> FROM Customers c
    -> LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
    -> GROUP BY c.CustomerID;
+-----------+-----------+------------+
| FirstName | LastName  | OrderCount |
+-----------+-----------+------------+
| Alice     | Johnson   |          1 |
| Bob       | Smith     |          3 |
| Carol     | Lee       |          0 |
| David     | Kumar     |          1 |
| Eva       | Rao       |          1 |
| Frank     | Das       |          2 |
| Grace     | Menon     |          1 |
| Henry     | Singh     |          1 |
| Ivy       | Patel     |          1 |
| John      | Fernandes |          0 |
| Ravi      | Sharma    |          1 |
+-----------+-----------+------------+
11 rows in set (0.00 sec)
```