HEXAWARE PYTHON BATCH -2

MySQL Assignment

S RUTH ROSHINI

TASK 1:

1. Create the database named "TechShop"

```
mysql> create database TechShop;
Query OK, 1 row affected (0.02 sec)

mysql> use TechShop;
Database changed
```

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

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

```
mysql> DESC Customers;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| CustomerID | int          | NO   | PRI | NULL    |       |
| FirstName  | varchar(50)  | YES  |     | NULL    |       |
| LastName   | varchar(50)  | YES  |     | NULL    |       |
| Email      | varchar(100) | YES  |     | NULL    |       |
| Phone      | varchar(20)  | YES  |     | NULL    |       |
| Address    | varchar(255) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

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

```
mysql> DESC Products;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| ProductID   | int           | NO   | PRI | NULL    |       |
| ProductName | varchar(100)  | YES  |     | NULL    |       |
| Description | text          | YES  |     | NULL    |       |
| Price       | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

```
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 (0.01 sec)
```

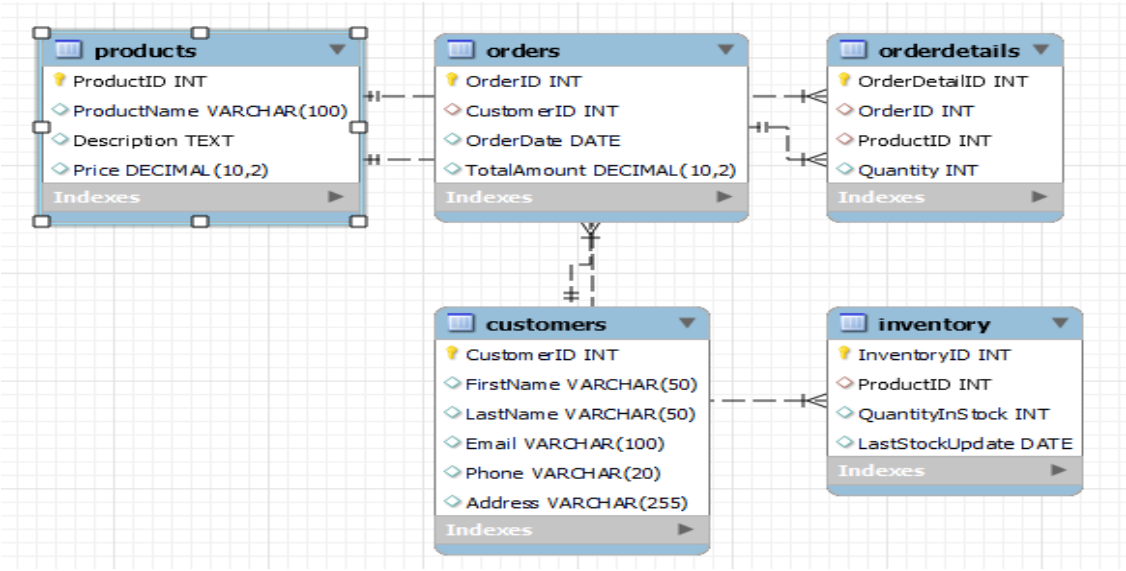
```
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

```
mysql> desc OrderDetails;
+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| OrderDetailID | int  | NO   | PRI | NULL    |       |
| OrderID       | int  | YES  | MUL | NULL    |       |
| ProductID     | int  | YES  | MUL | NULL    |       |
| Quantity      | int  | YES  |     | NULL    |       |
+---------------+------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

```
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate DATE,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

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

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



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

a. Customers :

```
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)
  VALUES (1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St'),
   (2, 'Jane', 'Smith', 'jane.smith@example.com', '987-654-3210', '456 Elm St'),
   (3, 'Michael', 'Johnson', 'michael.johnson@example.com', '111-222-3333', '789 Oak St'),
```

(4, 'Emily', 'Brown', 'emily.brown@example.com', '444-555-6666', '101 Pine St'),
(5, 'Chris', 'Davis', 'chris.davis@example.com', '777-888-9999', '202 Maple St'),
(6, 'Sarah', 'Wilson', 'sarah.wilson@example.com', '333-444-5555', '303 Cedar St'),
(7, 'David', 'Martinez', 'david.martinez@example.com', '666-777-8888', '404 Birch St'),
(8, 'Jessica', 'Taylor', 'jessica.taylor@example.com', '999-000-1111', '505 Walnut St'),
(9, 'Andrew', 'Anderson', 'andrew.anderson@example.com', '222-333-4444', '606 Pineapple St'),
(10, 'Laura', 'Lee', 'laura.lee@example.com', '555-666-7777', '707 Peach St');

```
mysql> select * from Customers;
+------------+-----------+----------+---------------------------+--------------+------------------+
| CustomerID | FirstName | LastName | Email                     | Phone        | Address          |
+------------+-----------+----------+---------------------------+--------------+------------------+
|          1 | John      | Doe      | john.doe@example.com      | 123-456-7890 | 123 Main St      |
|          2 | Jane      | Smith    | jane.smith@example.com    | 987-654-3210 | 456 Elm St       |
|          3 | Michael   | Johnson  | michael.johnson@example.com | 111-222-3333 | 789 Oak St     |
|          4 | Emily     | Brown    | emily.brown@example.com   | 444-555-6666 | 101 Pine St      |
|          5 | Chris     | Davis    | chris.davis@example.com   | 777-888-9999 | 202 Maple St     |
|          6 | Sarah     | Wilson   | sarah.wilson@example.com  | 333-444-5555 | 303 Cedar St     |
|          7 | David     | Martinez | david.martinez@example.com | 666-777-8888 | 404 Birch St    |
|          8 | Jessica   | Taylor   | jessica.taylor@example.com | 999-000-1111 | 505 Walnut St    |
|          9 | Andrew    | Anderson | andrew.anderson@example.com | 222-333-4444 | 606 Pineapple St |
|         10 | Laura     | Lee      | laura.lee@example.com     | 555-666-7777 | 707 Peach St     |
+------------+-----------+----------+---------------------------+--------------+------------------+
10 rows in set (0.00 sec)
```

b .Products:

INSERT INTO Products (ProductID, ProductName, Description, Price)
VALUES
(1, 'Laptop', '15-inch, Intel Core i5, 8GB RAM, 256GB SSD', 999.99),
(2, 'Smartphone', '6.5-inch, 128GB, Dual Camera, Android OS', 599.99),
(3, 'Tablet', '10-inch, 64GB, Wi-Fi, Touchscreen', 349.99),
(4, 'Headphones', 'Noise-canceling, Over-ear, Bluetooth', 149.99),
(5, 'Smartwatch', 'Water-resistant, Fitness Tracker, Heart Rate Monitor', 199.99),
(6, 'Speaker', 'Wireless, Bluetooth, 20W Output', 79.99),
(7, 'Monitor', '27-inch, Full HD, HDMI, VGA', 299.99),
(8, 'Keyboard', 'Mechanical, RGB Backlit, USB Wired', 69.99),
(9, 'Mouse', 'Wireless, Optical, Ergonomic Design', 29.99),
(10, 'External Hard Drive', '2TB, USB 3.0, Portable', 89.99);

```
mysql> select * from Products;
+-----------+--------------------+-----------------------------------------------------+--------+
| ProductID | ProductName        | Description                                         | Price  |
+-----------+--------------------+-----------------------------------------------------+--------+
|         1 | Laptop             | 15-inch, Intel Core i5, 8GB RAM, 256GB SSD          | 999.99 |
|         2 | Smartphone         | 6.5-inch, 128GB, Dual Camera, Android OS            | 599.99 |
|         3 | Tablet             | 10-inch, 64GB, Wi-Fi, Touchscreen                   | 349.99 |
|         4 | Headphones         | Noise-canceling, Over-ear, Bluetooth                | 149.99 |
|         5 | Smartwatch         | Water-resistant, Fitness Tracker, Heart Rate Monitor | 199.99 |
|         6 | Speaker            | Wireless, Bluetooth, 20W Output                     |  79.99 |
|         7 | Monitor            | 27-inch, Full HD, HDMI, VGA                         | 299.99 |
|         8 | Keyboard           | Mechanical, RGB Backlit, USB Wired                  |  69.99 |
|         9 | Mouse              | Wireless, Optical, Ergonomic Design                 |  29.99 |
|        10 | External Hard Drive | 2TB, USB 3.0, Portable                             |  89.99 |
+-----------+--------------------+-----------------------------------------------------+--------+
10 rows in set (0.00 sec)
```

c.Orders:

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
    VALUES
    (1, 1, '2024-04-01', 1099.98),
    (2, 2, '2024-04-02', 274.97),
    (3, 3, '2024-04-03', 679.96),
    (4, 4, '2024-04-04', 429.98),
    (5, 5, '2024-04-05', 749.95),
    (6, 6, '2024-04-06', 159.98),
    (7, 7, '2024-04-07', 149.99),
    (8, 8, '2024-04-08', 499.97),
    (9, 9, '2024-04-09', 1049.94),
    (10, 10, '2024-04-10', 279.97);
```

```
mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |          1 | 2024-04-01 |     1099.98 |
|       2 |          2 | 2024-04-02 |      274.97 |
|       3 |          3 | 2024-04-03 |      679.96 |
|       4 |          4 | 2024-04-04 |      429.98 |
|       5 |          5 | 2024-04-05 |      749.95 |
|       6 |          6 | 2024-04-06 |      159.98 |
|       7 |          7 | 2024-04-07 |      149.99 |
|       8 |          8 | 2024-04-08 |      499.97 |
|       9 |          9 | 2024-04-09 |     1049.94 |
|      10 |         10 | 2024-04-10 |      279.97 |
+---------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

d. Order Details:

```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
    VALUES
    (1, 1, 1, 1),
    (2, 1, 4, 1),
    (3, 2, 3, 2),
    (4, 3, 2, 1),
    (5, 3, 5, 1),
    (6, 3, 6, 1),
    (7, 4, 1, 1),
    (8, 5, 2, 1),
    (9, 5, 7, 1),
    (10, 6, 3, 1);
```

```
mysql> select * from OrderDetails;
+--------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+--------------+---------+-----------+----------+
|            1 |       1 |         1 |        1 |
|            2 |       1 |         4 |        1 |
|            3 |       2 |         3 |        2 |
|            4 |       3 |         2 |        1 |
|            5 |       3 |         5 |        1 |
|            6 |       3 |         6 |        1 |
|            7 |       4 |         1 |        1 |
|            8 |       5 |         2 |        1 |
|            9 |       5 |         7 |        1 |
|           10 |       6 |         3 |        1 |
+--------------+---------+-----------+----------+
10 rows in set (0.00 sec)
```

e. Inventory:

INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)

VALUES

(1, 1, 20, '2024-04-01'),

(2, 2, 15, '2024-04-02'),

(3, 3, 30, '2024-04-03'),

(4, 4, 25, '2024-04-04'),

(5, 5, 20, '2024-04-05'),

(6, 6, 40, '2024-04-06'),

(7, 7, 10, '2024-04-07'),

(8, 8, 35, '2024-04-08'),

(9, 9, 50, '2024-04-09'),

(10, 10, 18, '2024-04-10');

```
mysql> select *from Inventory;
+-------------+-----------+----------------+-----------------+
| InventoryID | ProductID | QuantityInStock | LastStockUpdate |
+-------------+-----------+----------------+-----------------+
|           1 |         1 |             20 | 2024-04-01      |
|           2 |         2 |             15 | 2024-04-02      |
|           3 |         3 |             30 | 2024-04-03      |
|           4 |         4 |             25 | 2024-04-04      |
|           5 |         5 |             20 | 2024-04-05      |
|           6 |         6 |             40 | 2024-04-06      |
|           7 |         7 |             10 | 2024-04-07      |
|           8 |         8 |             35 | 2024-04-08      |
|           9 |         9 |             50 | 2024-04-09      |
|          10 |        10 |             18 | 2024-04-10      |
+-------------+-----------+----------------+-----------------+
10 rows in set (0.00 sec)
```

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

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

   select FirstName ,LastName,Email from Customers;

```
mysql> select FirstName ,LastName,Email from Customers;
+-----------+----------+-----------------------------+
| FirstName | LastName | Email                       |
+-----------+----------+-----------------------------+
| John      | Doe      | john.doe@example.com        |
| Jane      | Smith    | jane.smith@example.com      |
| Michael   | Johnson  | michael.johnson@example.com |
| Emily     | Brown    | emily.brown@example.com     |
| Chris     | Davis    | chris.davis@example.com     |
| Sarah     | Wilson   | sarah.wilson@example.com    |
| David     | Martinez | david.martinez@example.com  |
| Jessica   | Taylor   | jessica.taylor@example.com  |
| Andrew    | Anderson | andrew.anderson@example.com |
| Laura     | Lee      | laura.lee@example.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.

   SELECT o.OrderID, o.OrderDate, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName
   FROM Orders o
   JOIN Customers c ON o.CustomerID = c.CustomerID;

```
mysql> SELECT o.OrderID, o.OrderDate, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName
    -> FROM Orders o
    -> JOIN Customers c ON o.CustomerID = c.CustomerID;
+---------+------------+----------------+
| OrderID | OrderDate  | CustomerName   |
+---------+------------+----------------+
|       1 | 2024-04-01 | John Doe       |
|       2 | 2024-04-02 | Jane Smith     |
|       3 | 2024-04-03 | Michael Johnson |
|       4 | 2024-04-04 | Emily Brown    |
|       5 | 2024-04-05 | Chris Davis    |
|       6 | 2024-04-06 | Sarah Wilson   |
|       7 | 2024-04-07 | David Martinez |
|       8 | 2024-04-08 | Jessica Taylor |
|       9 | 2024-04-09 | Andrew Anderson |
|      10 | 2024-04-10 | Laura Lee      |
+---------+------------+----------------+
10 rows in set (0.00 sec)
```

3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.

   INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address) VALUES (11, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St');

```
mysql> select * from Customers;
+------------+-----------+----------+---------------------------+-------------+-----------------+
| CustomerID | FirstName | LastName | Email                     | Phone       | Address         |
+------------+-----------+----------+---------------------------+-------------+-----------------+
|          1 | John      | Doe      | john.doe@example.com      | 123-456-7890 | 123 Main St     |
|          2 | Jane      | Smith    | jane.smith@example.com    | 987-654-3210 | 456 Elm St      |
|          3 | Michael   | Johnson  | michael.johnson@example.com | 111-222-3333 | 789 Oak St    |
|          4 | Emily     | Brown    | emily.brown@example.com   | 444-555-6666 | 101 Pine St     |
|          5 | Chris     | Davis    | chris.davis@example.com   | 777-888-9999 | 202 Maple St    |
|          6 | Sarah     | Wilson   | sarah.wilson@example.com  | 333-444-5555 | 303 Cedar St    |
|          7 | David     | Martinez | david.martinez@example.com | 666-777-8888 | 404 Birch St   |
|          8 | Jessica   | Taylor   | jessica.taylor@example.com | 999-000-1111 | 505 Walnut St  |
|          9 | Andrew    | Anderson | andrew.anderson@example.com | 222-333-4444 | 606 Pineapple St |
|         10 | Laura     | Lee      | laura.lee@example.com     | 555-666-7777 | 707 Peach St    |
|         11 | John      | Doe      | john.doe@example.com      | 123-456-7890 | 123 Main St     |
+------------+-----------+----------+---------------------------+-------------+-----------------+
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%.

   UPDATE Products SET Price = Price * 1.10;

```
mysql> select * from Products;
+-----------+--------------------+-----------------------------------------------------------+---------+
| ProductID | ProductName        | Description                                               | Price   |
+-----------+--------------------+-----------------------------------------------------------+---------+
|         1 | Laptop             | 15-inch, Intel Core i5, 8GB RAM, 256GB SSD                | 1099.99 |
|         2 | Smartphone         | 6.5-inch, 128GB, Dual Camera, Android OS                  |  659.99 |
|         3 | Tablet             | 10-inch, 64GB, Wi-Fi, Touchscreen                         |  384.99 |
|         4 | Headphones         | Noise-canceling, Over-ear, Bluetooth                      |  164.99 |
|         5 | Smartwatch         | Water-resistant, Fitness Tracker, Heart Rate Monitor      |  219.99 |
|         6 | Speaker            | Wireless, Bluetooth, 20W Output                           |   87.99 |
|         7 | Monitor            | 27-inch, Full HD, HDMI, VGA                               |  329.99 |
|         8 | Keyboard           | Mechanical, RGB Backlit, USB Wired                        |   76.99 |
|         9 | Mouse              | Wireless, Optical, Ergonomic Design                       |   32.99 |
|        10 | External Hard Drive | 2TB, USB 3.0, Portable                                   |   98.99 |
+-----------+--------------------+-----------------------------------------------------------+---------+
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.

   DELETE FROM OrderDetails WHERE OrderID = 3;

```
mysql> SELECT * FROM OrderDetails;
+---------------+---------+-----------+----------+
| OrderDetailID | OrderID | ProductID | Quantity |
+---------------+---------+-----------+----------+
|             1 |       1 |         1 |        1 |
|             2 |       1 |         4 |        1 |
|             3 |       2 |         3 |        2 |
|             7 |       4 |         1 |        1 |
|             8 |       5 |         2 |        1 |
|             9 |       5 |         7 |        1 |
|            10 |       6 |         3 |        1 |
+---------------+---------+-----------+----------+
7 rows in set (0.01 sec)
```

6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
    -> VALUES (11, 11, '2024-04-11', 158.0);
```

```
mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |          1 | 2024-04-01 |     1099.98 |
|       2 |          2 | 2024-04-02 |      274.97 |
|       3 |          3 | 2024-04-03 |      679.96 |
|       4 |          4 | 2024-04-04 |      429.98 |
|       5 |          5 | 2024-04-05 |      749.95 |
|       6 |          6 | 2024-04-06 |      159.98 |
|       7 |          7 | 2024-04-07 |      149.99 |
|       8 |          8 | 2024-04-08 |      499.97 |
|       9 |          9 | 2024-04-09 |     1049.94 |
|      10 |         10 | 2024-04-10 |      279.97 |
|      11 |         11 | 2024-04-11 |      158.00 |
+---------+------------+------------+-------------+
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.

```
UPDATE Customers
SET Email = 'sony.email@example.com', Address = '456 Elm St'
where CustomerID=11;
```

```
mysql> select * from Customers;
+------------+-----------+----------+----------------------------+--------------+------------------+
| CustomerID | FirstName | LastName | Email                      | Phone        | Address          |
+------------+-----------+----------+----------------------------+--------------+------------------+
|          1 | John      | Doe      | john.doe@example.com       | 123-456-7890 | 123 Main St      |
|          2 | Jane      | Smith    | jane.smith@example.com     | 987-654-3210 | 456 Elm St       |
|          3 | Michael   | Johnson  | michael.johnson@example.com| 111-222-3333 | 789 Oak St       |
|          4 | Emily     | Brown    | emily.brown@example.com    | 444-555-6666 | 101 Pine St      |
|          5 | Chris     | Davis    | chris.davis@example.com    | 777-888-9999 | 202 Maple St     |
|          6 | Sarah     | Wilson   | sarah.wilson@example.com   | 333-444-5555 | 303 Cedar St     |
|          7 | David     | Martinez | david.martinez@example.com | 666-777-8888 | 404 Birch St     |
|          8 | Jessica   | Taylor   | jessica.taylor@example.com | 999-000-1111 | 505 Walnut St    |
|          9 | Andrew    | Anderson | andrew.anderson@example.com| 222-333-4444 | 606 Pineapple St |
|         10 | Laura     | Lee      | laura.lee@example.com      | 555-666-7777 | 707 Peach St     |
|         11 | Sony      | Ray      | sony.email@example.com     | 123-456-7890 | 456 Elm St       |
+------------+-----------+----------+----------------------------+--------------+------------------+
11 rows in set (0.00 sec)
```

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

```
UPDATE Orders AS o
    SET TotalAmount = (
       SELECT SUM(od.Quantity * p.Price) FROM OrderDetails od
       JOIN Products p ON od.ProductID = p.ProductID
      WHERE od.OrderID = o.OrderID )
       WHERE EXISTS ( SELECT 1  FROM OrderDetails od
     WHERE od.OrderID = o.OrderID );
```

```
mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |          1 | 2024-04-01 |     1264.98 |
|       2 |          2 | 2024-04-02 |      769.98 |
|       3 |          3 | 2024-04-03 |      679.96 |
|       4 |          4 | 2024-04-04 |     1099.99 |
|       5 |          5 | 2024-04-05 |      989.98 |
|       6 |          6 | 2024-04-06 |      384.99 |
|       7 |          7 | 2024-04-07 |      149.99 |
|       8 |          8 | 2024-04-08 |      499.97 |
|       9 |          9 | 2024-04-09 |     1049.94 |
|      10 |         10 | 2024-04-10 |      279.97 |
|      11 |         11 | 2024-04-11 |      158.00 |
+---------+------------+------------+-------------+
11 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.

```
DELETE FROM Orders

WHERE CustomerID = 11;
```

```
mysql> select * from Orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |          1 | 2024-04-01 |     1264.98 |
|       2 |          2 | 2024-04-02 |      769.98 |
|       3 |          3 | 2024-04-03 |      679.96 |
|       4 |          4 | 2024-04-04 |     1099.99 |
|       5 |          5 | 2024-04-05 |      989.98 |
|       6 |          6 | 2024-04-06 |      384.99 |
|       7 |          7 | 2024-04-07 |      149.99 |
|       8 |          8 | 2024-04-08 |      499.97 |
|       9 |          9 | 2024-04-09 |     1049.94 |
|      10 |         10 | 2024-04-10 |      279.97 |
+---------+------------+------------+-------------+
10 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.

    INSERT INTO Products (ProductID, ProductName, Description, Price)
    VALUES ('11', 'Television', 'Smart, HD resolution', 45999.99);

```
mysql> select * from Products;
+-----------+--------------------+-----------------------------------------------------+----------+
| ProductID | ProductName        | Description                                         | Price    |
+-----------+--------------------+-----------------------------------------------------+----------+
|         1 | Laptop             | 15-inch, Intel Core i5, 8GB RAM, 256GB SSD          |  1099.99 |
|         2 | Smartphone         | 6.5-inch, 128GB, Dual Camera, Android OS            |   659.99 |
|         3 | Tablet             | 10-inch, 64GB, Wi-Fi, Touchscreen                   |   384.99 |
|         4 | Headphones         | Noise-canceling, Over-ear, Bluetooth                |   164.99 |
|         5 | Smartwatch         | Water-resistant, Fitness Tracker, Heart Rate Monitor |   219.99 |
|         6 | Speaker            | Wireless, Bluetooth, 20W Output                     |    87.99 |
|         7 | Monitor            | 27-inch, Full HD, HDMI, VGA                         |   329.99 |
|         8 | Keyboard           | Mechanical, RGB Backlit, USB Wired                  |    76.99 |
|         9 | Mouse              | Wireless, Optical, Ergonomic Design                 |    32.99 |
|        10 | External Hard Drive | 2TB, USB 3.0, Portable                              |    98.99 |
|        11 | Television         | Smart, HD resolution                                | 45999.99 |
+-----------+--------------------+-----------------------------------------------------+----------+
11 rows in set (0.01 sec)
```

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

```
mysql> select * from Orders;
+---------+------------+------------+-------------+---------+
| OrderID | CustomerID | OrderDate  | TotalAmount | Status  |
+---------+------------+------------+-------------+---------+
|       1 |          1 | 2024-04-01 |     1264.98 | Shipped |
|       2 |          2 | 2024-04-02 |      769.98 | Shipped |
|       3 |          3 | 2024-04-03 |      679.96 | Shipped |
|       4 |          4 | 2024-04-04 |     1099.99 | Shipped |
|       5 |          5 | 2024-04-05 |      989.98 | Shipped |
|       6 |          6 | 2024-04-06 |      384.99 | Shipped |
|       7 |          7 | 2024-04-07 |      149.99 | Shipped |
|       8 |          8 | 2024-04-08 |      499.97 | Shipped |
|       9 |          9 | 2024-04-09 |     1049.94 | Shipped |
|      10 |         10 | 2024-04-10 |      279.97 | Shipped |
+---------+------------+------------+-------------+---------+
10 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.

    ALTER TABLE Customers
    ADD COLUMN TotalOrders INT;

```
mysql> desc Customers;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| CustomerID  | int          | NO   | PRI | NULL    |       |
| FirstName   | varchar(50)  | YES  |     | NULL    |       |
| LastName    | varchar(50)  | YES  |     | NULL    |       |
| Email       | varchar(100) | YES  |     | NULL    |       |
| Phone       | varchar(20)  | YES  |     | NULL    |       |
| Address     | varchar(255) | YES  |     | NULL    |       |
| TotalOrders | int          | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

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

```
mysql> select * from Customers;
+------------+-----------+----------+-----------------------------+--------------+------------------+-------------+
| CustomerID | FirstName | LastName | Email                       | Phone        | Address          | TotalOrders |
+------------+-----------+----------+-----------------------------+--------------+------------------+-------------+
|          1 | John      | Doe      | john.doe@example.com        | 123-456-7890 | 123 Main St      |           1 |
|          2 | Jane      | Smith    | jane.smith@example.com      | 987-654-3210 | 456 Elm St       |           1 |
|          3 | Michael   | Johnson  | michael.johnson@example.com | 111-222-3333 | 789 Oak St       |           1 |
|          4 | Emily     | Brown    | emily.brown@example.com     | 444-555-6666 | 101 Pine St      |           1 |
|          5 | Chris     | Davis    | chris.davis@example.com     | 777-888-9999 | 202 Maple St     |           1 |
|          6 | Sarah     | Wilson   | sarah.wilson@example.com    | 333-444-5555 | 303 Cedar St     |           1 |
|          7 | David     | Martinez | david.martinez@example.com  | 666-777-8888 | 404 Birch St     |           1 |
|          8 | Jessica   | Taylor   | jessica.taylor@example.com  | 999-000-1111 | 505 Walnut St    |           1 |
|          9 | Andrew    | Anderson | andrew.anderson@example.com | 222-333-4444 | 606 Pineapple St |           1 |
|         10 | Laura     | Lee      | laura.lee@example.com       | 555-666-7777 | 707 Peach St     |           1 |
|         11 | Sony      | Ray      | sony.email@example.com      | 123-456-7890 | 456 Elm St       |           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.

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

OrderID	OrderDate	FirstName	LastName
1	2024-04-01	John	Doe
2	2024-04-02	Jane	Smith
3	2024-04-03	Michael	Johnson
4	2024-04-04	Emily	Brown
5	2024-04-05	Chris	Davis
6	2024-04-06	Sarah	Wilson
7	2024-04-07	David	Martinez
8	2024-04-08	Jessica	Taylor
9	2024-04-09	Andrew	Anderson
10	2024-04-10	Laura	Lee

10 rows in set (0.11 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.

```
SELECT p.ProductName, SUM(o.TotalAmount) AS TotalRevenue
FROM Orders o
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName;
```

ProductName	TotalRevenue
Laptop	2364.97
Headphones	1264.98
Tablet	1154.97
Smartphone	989.98
Monitor	989.98

5 rows in set (0.07 sec)

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

```
SELECT FirstName, LastName, Email, Phone
 FROM Customers
WHERE CustomerID IN (SELECT DISTINCT CustomerID FROM Orders);
```

FirstName	LastName	Email	Phone
John	Doe	john.doe@example.com	123-456-7890
Jane	Smith	jane.smith@example.com	987-654-3210
Michael	Johnson	michael.johnson@example.com	111-222-3333
Emily	Brown	emily.brown@example.com	444-555-6666
Chris	Davis	chris.davis@example.com	777-888-9999
Sarah	Wilson	sarah.wilson@example.com	333-444-5555
David	Martinez	david.martinez@example.com	666-777-8888
Jessica	Taylor	jessica.taylor@example.com	999-000-1111
Andrew	Anderson	andrew.anderson@example.com	222-333-4444
Laura	Lee	laura.lee@example.com	555-666-7777

10 rows in set (0.01 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.

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

ProductName	TotalQuantityOrdered
Tablet	3

1 row in set (0.01 sec)

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

```
ALTER TABLE Products
ADD COLUMN Category VARCHAR(50);

UPDATE Products
SET Category = CASE
  WHEN ProductID IN (1, 2, 3) THEN 'Electronics'
  WHEN ProductID IN (4, 5, 6) THEN 'Accessories'
  WHEN ProductID IN (7, 8, 9, 10) THEN 'Peripherals'
  WHEN ProductID = 11 THEN 'Electronics'
  ELSE 'Unknown'
END;

SELECT ProductName, Category
FROM Products;
```

ProductName	Category
Laptop	Electronics
Smartphone	Electronics
Tablet	Electronics
Headphones	Accessories
Smartwatch	Accessories
Speaker	Accessories
Monitor	Peripherals
Keyboard	Peripherals
Mouse	Peripherals
External Hard Drive	Peripherals
Television	Electronics

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

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

FirstName	LastName	AverageOrderValue
John	Doe	1264.980000
Jane	Smith	769.980000
Michael	Johnson	679.960000
Emily	Brown	1099.990000
Chris	Davis	989.980000
Sarah	Wilson	384.990000
David	Martinez	149.990000
Jessica	Taylor	499.970000
Andrew	Anderson	1049.940000
Laura	Lee	279.970000

10 rows in set (0.01 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.

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

```
+---------+-----------+----------+--------------+
| OrderID | FirstName | LastName | TotalRevenue |
+---------+-----------+----------+--------------+
|       1 | John      | Doe      |      1264.98 |
+---------+-----------+----------+--------------+
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.

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

```
+-------------+------------+
| ProductName | OrderCount |
+-------------+------------+
| Laptop      |          2 |
| Smartphone  |          1 |
| Tablet      |          2 |
| Headphones  |          1 |
| Monitor     |          1 |
+-------------+------------+
5 rows in set (0.01 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

```
SELECT 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
John	Doe	john.doe@example.com	123-456-7890
Emily	Brown	emily.brown@example.com	444-555-6666

2 rows 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.

```
SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders
WHERE OrderDate BETWEEN 2024-04-03 AND 2024-04-10;
```

TotalRevenue
NULL

1 row in set, 2 warnings (0.02 sec)

Task 4. Subquery and its type:

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

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

FirstName	LastName	Email	Phone
Sony	Ray	sony.email@example.com	123-456-7890

1 row in set (0.01 sec)

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

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

TotalProducts
11

1 row in set (0.02 sec)

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

```
SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders;
```

```
+--------------+
| TotalRevenue |
+--------------+
|      7169.75 |
+--------------+
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.

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

```
+------------------------+
| AverageQuantityOrdered |
+------------------------+
|                 1.2000 |
+------------------------+
1 row in set (0.00 sec)
```

5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.

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

```
+--------------+
| TotalRevenue |
+--------------+
|      1264.98 |
+--------------+
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.

```
SELECT c.FirstName, c.LastName, COUNT(o.OrderID) AS OrderCount
FROM Customers c
```

JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID
ORDER BY OrderCount DESC
LIMIT 1;

```
+-----------+----------+------------+
| FirstName | LastName | OrderCount |
+-----------+----------+------------+
| John      | Doe      |          1 |
+-----------+----------+------------+
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.

SELECT p.Category, SUM(od.Quantity) AS TotalQuantityOrdered

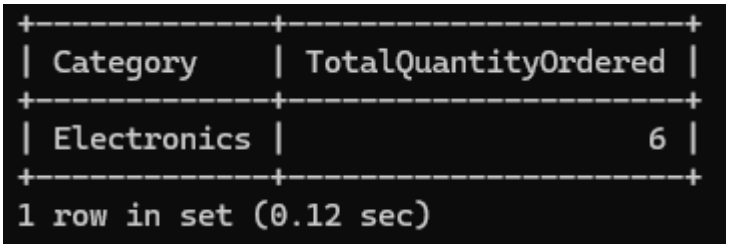
FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY p.Category

ORDER BY TotalQuantityOrdered DESC

LIMIT 1;



```
+-------------+----------------------+
| Category    | TotalQuantityOrdered |
+-------------+----------------------+
| Electronics |                    6 |
+-------------+----------------------+
1 row in set (0.12 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.

SELECT c.FirstName, c.LastName, SUM(o.TotalAmount) AS TotalSpending

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

GROUP BY c.CustomerID

ORDER BY TotalSpending DESC

LIMIT 1;

FirstName	LastName	TotalSpending
John	Doe	1264.98

1 row in set (0.05 sec)

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

```
SELECT AVG(TotalAmount) AS AverageOrderValue

FROM Orders;
```

AverageOrderValue
716.975000

1 row in set (0.00 sec)

10. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.

```
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
John	Doe	1
Jane	Smith	1
Michael	Johnson	1
Emily	Brown	1
Chris	Davis	1
Sarah	Wilson	1
David	Martinez	1
Jessica	Taylor	1
Andrew	Anderson	1
Laura	Lee	1
Sony	Ray	0

11 rows in set (0.00 sec)