2100030748

BACKEND

→Firstly open myql and create a database

mysql> create database retail_store;

Query OK, 1 row affected (0.02 sec)

→Change the database which was newly created.

mysql> use retail_store;

Database changed

→Now **create tables** and Insert the data into the tables:

1)Customers:

CustomerID	FirstName	LastName	Email	DateOfBirth
1	John	Doe	john.doe@example.com	1985-01-15
2	Jane	Smith	jane.smith@example.com	1990-06-20

```
mysql> CREATE TABLE Customers (
    ->     CustomerID INT PRIMARY KEY,
    ->     FirstName VARCHAR(50),
    ->     LastName VARCHAR(50),
    ->     Email VARCHAR(100),
    ->     DateOfBirth DATE
    -> );
Query OK, 0 rows affected (0.10 sec)
```

mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, DateOfBirth) VALUES

-> (1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'),

-> (2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');

Query OK, 2 rows affected (0.02 sec)

Records: 2 Duplicates: 0 Warnings: 0

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

mysql> use retail_store;
Database changed
mysql> CREATE TABLE Customers (
    ->     CustomerID INT PRIMARY KEY,
    ->     FirstName VARCHAR(50),
    ->     LastName VARCHAR(50),
    ->     Email VARCHAR(100),
    ->     DateOfBirth DATE
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, DateOfBirth) VALUES
    -> (1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'),
    -> (2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');
Query OK, 2 rows affected (0.02 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

2)Products:

ProductID	ProductName	Price
1	Laptop	1000
2	Smartphone	600
3	Headphones	100

mysql> CREATE TABLE Products (

-> ProductID INT PRIMARY KEY,

-> ProductName VARCHAR(100),

-> Price DECIMAL(10, 2)

->);

Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO Products (ProductID, ProductName, Price) VALUES

-> (1, 'Laptop', 1000),

-> (2, 'Smartphone', 600),

-> (3, 'Headphones', 100);

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

```
mysql> CREATE TABLE Products (
    ->     ProductID INT PRIMARY KEY,
    ->     ProductName VARCHAR(100),
    ->     Price DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO Products (ProductID, ProductName, Price) VALUES
    -> (1, 'Laptop', 1000),
    -> (2, 'Smartphone', 600),
    -> (3, 'Headphones', 100);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

3)Orders:

OrderID	CustomerID	OrderDate
1	1	2023-01-10
2	2	2023-01-12

mysql> CREATE TABLE Orders (

-> OrderID INT PRIMARY KEY,

-> CustomerID INT,

-> OrderDate DATE,

-> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

->);

Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES

-> (1, 1, '2023-01-10'),

-> (2, 2, '2023-01-12');

Query OK, 2 rows affected (0.01 sec)

Records: 2 Duplicates: 0 Warnings: 0

```
mysql> CREATE TABLE Orders (
    ->     OrderID INT PRIMARY KEY,
    ->     CustomerID INT,
    ->     OrderDate DATE,
    ->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES
    -> (1, 1, '2023-01-10'),
    -> (2, 2, '2023-01-12');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

4) OrderItems:

OrderItemID	OrderID	ProductID	Quantity
1	1	1	1
2	1	3	2
3	2	2	1
4	2	3	1

mysql> CREATE TABLE OrderItems (

-> OrderItemID INT PRIMARY KEY,

-> OrderID INT,

-> ProductID INT,

-> Quantity INT,

-> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

-> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

->);

Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity) VALUES

-> (1, 1, 1, 1),

-> (2, 1, 3, 2),

-> (3, 2, 2, 1),

-> (4, 2, 3, 1);

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates: 0 Warnings: 0

```
mysql> CREATE TABLE OrderItems (
    ->     OrderItemID INT PRIMARY KEY,
    ->     OrderID INT,
    ->     ProductID INT,
    ->     Quantity INT,
    ->     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
    -> );
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity) VALUES
    -> (1, 1, 1, 1),
    -> (2, 1, 3, 2),
    -> (3, 2, 2, 1),
    -> (4, 2, 3, 1);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Now we have created all 4 tables and inserted data into the tables successfully:

We have queries to perform on above tables:

1.List all customers.

mysql> select * from Customers;

```
mysql> select * from Customers;
+------------+-----------+----------+-----------------------+-------------+
| CustomerID | FirstName | LastName | Email                 | DateOfBirth |
+------------+-----------+----------+-----------------------+-------------+
|          1 | John      | Doe      | john.doe@example.com  | 1985-01-15  |
|          2 | Jane      | Smith    | jane.smith@example.com | 1990-06-20  |
+------------+-----------+----------+-----------------------+-------------+
2 rows in set (0.01 sec)
```

2. Find all orders placed in January 2023.

mysql> SELECT * FROM Orders

-> WHERE OrderDate BETWEEN '2023-01-01' AND '2023-01-31';

```
mysql> SELECT * FROM Orders
    -> WHERE OrderDate BETWEEN '2023-01-01' AND '2023-01-31';
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|       1 |          1 | 2023-01-10 |
|       2 |          2 | 2023-01-12 |
+---------+------------+------------+
2 rows in set (0.00 sec)
```

3. Get the details of each order, including the customer name and email.

mysql> SELECT Orders.OrderID, Customers.FirstName, Customers.LastName, Customers.Email, Orders.OrderDate

-> FROM Orders

-> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

```
mysql> SELECT Orders.OrderID, Customers.FirstName, Customers.LastName, Customers.Email, Orders.OrderDate
    -> FROM Orders
    -> JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
+---------+-----------+----------+-----------------------+------------+
| OrderID | FirstName | LastName | Email                 | OrderDate  |
+---------+-----------+----------+-----------------------+------------+
|       1 | John      | Doe      | john.doe@example.com  | 2023-01-10 |
|       2 | Jane      | Smith    | jane.smith@example.com | 2023-01-12 |
+---------+-----------+----------+-----------------------+------------+
2 rows in set (0.00 sec)
```

4. List the products purchased in a specific order (e.g., OrderID = 1).

mysql> SELECT Products.ProductName, OrderItems.Quantity

-> FROM OrderItems

-> JOIN Products ON OrderItems.ProductID = Products.ProductID

-> WHERE OrderItems.OrderID = 1;

```
mysql> SELECT Products.ProductName, OrderItems.Quantity
    -> FROM OrderItems
    -> JOIN Products ON OrderItems.ProductID = Products.ProductID
    -> WHERE OrderItems.OrderID = 1;
+-------------+----------+
| ProductName | Quantity |
+-------------+----------+
| Laptop      |        1 |
| Headphones  |        2 |
+-------------+----------+
2 rows in set (0.01 sec)
```

5. Calculate the total amount spent by each customer.

mysql> SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName, SUM(Products.Price * OrderItems.Quantity) AS TotalSpent

-> FROM Customers

-> JOIN Orders ON Customers.CustomerID = Orders.CustomerID

-> JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID

-> JOIN Products ON OrderItems.ProductID = Products.ProductID

-> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName;

```
mysql> SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName, SUM(Products.Price * OrderItems.Quantity) AS TotalSpent
    -> FROM Customers
    -> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID
    -> JOIN Products ON OrderItems.ProductID = Products.ProductID
    -> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName;
+------------+-----------+----------+------------+
| CustomerID | FirstName | LastName | TotalSpent |
+------------+-----------+----------+------------+
|          1 | John      | Doe      |    1200.00 |
|          2 | Jane      | Smith    |     700.00 |
+------------+-----------+----------+------------+
2 rows in set (0.01 sec)
```

6. Find the most popular product (the one that has been ordered the most).

mysql> SELECT Products.ProductID, Products.ProductName, SUM(OrderItems.Quantity) AS TotalQuantity

-> FROM OrderItems

-> JOIN Products ON OrderItems.ProductID = Products.ProductID

-> GROUP BY Products.ProductID, Products.ProductName

-> ORDER BY TotalQuantity DESC

-> LIMIT 1;

```
mysql> SELECT Products.ProductID, Products.ProductName, SUM(OrderItems.Quantity) AS TotalQuantity
    -> FROM OrderItems
    -> JOIN Products ON OrderItems.ProductID = Products.ProductID
    -> GROUP BY Products.ProductID, Products.ProductName
    -> ORDER BY TotalQuantity DESC
    -> LIMIT 1;
+-----------+-------------+---------------+
| ProductID | ProductName | TotalQuantity |
+-----------+-------------+---------------+
|         3 | Headphones  |             3 |
+-----------+-------------+---------------+
1 row in set (0.00 sec)
```

7. Get the total number of orders and the total sales amount for each month in 2023.

mysql> SELECT DATE_FORMAT(Orders.OrderDate, '%Y-%m') AS Month,

-> COUNT(Orders.OrderID) AS TotalOrders,

-> SUM(Products.Price * OrderItems.Quantity) AS TotalSales

-> FROM Orders

-> JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID

-> JOIN Products ON OrderItems.ProductID = Products.ProductID

-> WHERE YEAR(Orders.OrderDate) = 2023

-> GROUP BY DATE_FORMAT(Orders.OrderDate, '%Y-%m');

```
mysql> SELECT DATE_FORMAT(Orders.OrderDate, '%Y-%m') AS Month,
    ->        COUNT(Orders.OrderID) AS TotalOrders,
    ->        SUM(Products.Price * OrderItems.Quantity) AS TotalSales
    -> FROM Orders
    -> JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID
    -> JOIN Products ON OrderItems.ProductID = Products.ProductID
    -> WHERE YEAR(Orders.OrderDate) = 2023
    -> GROUP BY DATE_FORMAT(Orders.OrderDate, '%Y-%m');
+---------+-------------+------------+
| Month   | TotalOrders | TotalSales |
+---------+-------------+------------+
| 2023-01 |           4 |    1900.00 |
+---------+-------------+------------+
1 row in set (0.00 sec)
```

8.Find customers who have spent more than $1000.

mysql> SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName, SUM(Products.Price * OrderItems.Quantity) AS TotalSpent

-> FROM Customers

-> JOIN Orders ON Customers.CustomerID = Orders.CustomerID

-> JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID

-> JOIN Products ON OrderItems.ProductID = Products.ProductID

-> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName

-> HAVING TotalSpent > 1000;

```
mysql> SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName, SUM(Products.Price * OrderItems.Quantity) AS TotalSpent
    -> FROM Customers
    -> JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID
    -> JOIN Products ON OrderItems.ProductID = Products.ProductID
    -> GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName
    -> HAVING TotalSpent > 1000;
+------------+-----------+----------+------------+
| CustomerID | FirstName | LastName | TotalSpent |
+------------+-----------+----------+------------+
|          1 | John      | Doe      |    1200.00 |
+------------+-----------+----------+------------+
1 row in set (0.00 sec)
```