

BACKEND TASK

N. Navya Sree

2100031266

```
MySQL 8.0 Command Line Cli. x + v
Copyright (c) 2000, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use backend;
Database changed
mysql> CREATE TABLE Customers (
  --> CustomerID INTEGER PRIMARY KEY,
  --> FirstName VARCHAR(30) NOT NULL,
  --> LastName VARCHAR(30) NOT NULL,
  --> Email VARCHAR(40) NOT NULL,
  --> DateOfBirth DATE NOT NULL,
  --> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Products (
  --> ProductID INTEGER PRIMARY KEY,
  --> ProductName VARCHAR(30) NOT NULL,
  --> Price INTEGER NOT NULL,
  --> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Orders (
  --> OrderID INTEGER PRIMARY KEY,
  --> CustomerID INTEGER NOT NULL,
  --> OrderDate DATE NOT NULL,
  --> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
  --> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE OrderItems (
  --> OrderItemID INTEGER PRIMARY KEY,
  --> OrderID INTEGER NOT NULL,
  --> ProductID INTEGER NOT NULL,
  --> Quantity INTEGER NOT NULL,
  --> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
  --> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
  --> );
Query OK, 0 rows affected (0.07 sec)

mysql>
mysql> INSERT INTO Customers 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>
mysql> INSERT INTO Customers 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> INSERT INTO Products VALUES
  --> (1, 'Laptop', 1000),
  --> (2, 'Smartphone', 600),
  --> (3, 'Headphones', 150);
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Orders VALUES
  --> (1, 1, '2023-01-10'),
  --> (2, 2, '2023-01-12');
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0

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> 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 (0.00 sec)

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

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

```
MySQL 8.0 Command Line CL  X  +  -

mysql> SELECT o.OrderID, c.FirstName, c.LastName, c.Email, o.OrderDate
-> FROM Orders o JOIN Customers c ON o.CustomerID = c.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)

mysql> SELECT oi.OrderID, oi.ProductID, p.ProductName, oi.Quantity
-> FROM OrderItems oi JOIN Products p ON oi.ProductID = p.ProductID
-> ORDER BY oi.OrderID ASC;

+-----+-----+-----+-----+
| OrderID | ProductID | ProductName | Quantity |
+-----+-----+-----+-----+
| 1 | 1 | Laptop | 1 |
| 1 | 3 | Headphones | 2 |
| 2 | 2 | Smartphone | 1 |
| 2 | 3 | Headphones | 1 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT c.CustomerID, c.FirstName, c.LastName, SUM(p.Price * oi.Quantity) AS TotalAmountSpent
-> FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID
-> JOIN OrderItems oi ON o.OrderID = oi.OrderID
-> JOIN Products p ON oi.ProductID = p.ProductID
-> GROUP BY c.CustomerID;

+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | TotalAmountSpent |
+-----+-----+-----+-----+
| 1 | John | Doe | 1200 |
| 2 | Jane | Smith | 700 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT p.ProductName, COUNT(oi.Quantity) AS TotalQuantity
-> FROM Products p JOIN OrderItems oi ON p.ProductID = oi.ProductID
-> GROUP BY p.ProductID ORDER BY TotalQuantity DESC;

+-----+-----+
| ProductName | TotalQuantity |
+-----+-----+
| Headphones | 2 |
| Laptop | 1 |
| Smartphone | 1 |
+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT MONTH(o.OrderDate) AS Month, COUNT(DISTINCT o.OrderID) AS TotalOrders,
-> SUM(p.Price * oi.Quantity) AS TotalSalesAmount FROM Orders o
```

```
MySQL 8.0 Command Line CL  X  +  -

4 rows in set (0.00 sec)

mysql> SELECT c.CustomerID, c.FirstName, c.LastName, SUM(p.Price * oi.Quantity) AS TotalAmountSpent
-> FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID
-> JOIN OrderItems oi ON o.OrderID = oi.OrderID
-> JOIN Products p ON oi.ProductID = p.ProductID
-> GROUP BY c.CustomerID;

+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | TotalAmountSpent |
+-----+-----+-----+-----+
| 1 | John | Doe | 1200 |
| 2 | Jane | Smith | 700 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT p.ProductName, COUNT(oi.Quantity) AS TotalQuantity
-> FROM Products p JOIN OrderItems oi ON p.ProductID = oi.ProductID
-> GROUP BY p.ProductID ORDER BY TotalQuantity DESC;

+-----+-----+
| ProductName | TotalQuantity |
+-----+-----+
| Headphones | 2 |
| Laptop | 1 |
| Smartphone | 1 |
+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT MONTH(o.OrderDate) AS Month, COUNT(DISTINCT o.OrderID) AS TotalOrders,
-> SUM(p.Price * oi.Quantity) AS TotalSalesAmount FROM Orders o
-> JOIN OrderItems oi ON o.OrderID = oi.OrderID
-> JOIN Products p ON oi.ProductID = p.ProductID
-> WHERE YEAR(o.OrderDate) = 2023 GROUP BY MONTH(o.OrderDate);

+-----+-----+-----+
| Month | TotalOrders | TotalSalesAmount |
+-----+-----+-----+
| 1 | 2 | 1900 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT c.CustomerID, c.FirstName, c.LastName, SUM(p.Price * oi.Quantity) AS TotalAmountSpent
-> FROM Customers c JOIN Orders o ON c.CustomerID = o.CustomerID
-> JOIN OrderItems oi ON o.OrderID = oi.OrderID
-> JOIN Products p ON oi.ProductID = p.ProductID
-> GROUP BY c.CustomerID HAVING SUM(p.Price * oi.Quantity) > 1000;

+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | TotalAmountSpent |
+-----+-----+-----+-----+
| 1 | John | Doe | 1200 |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```