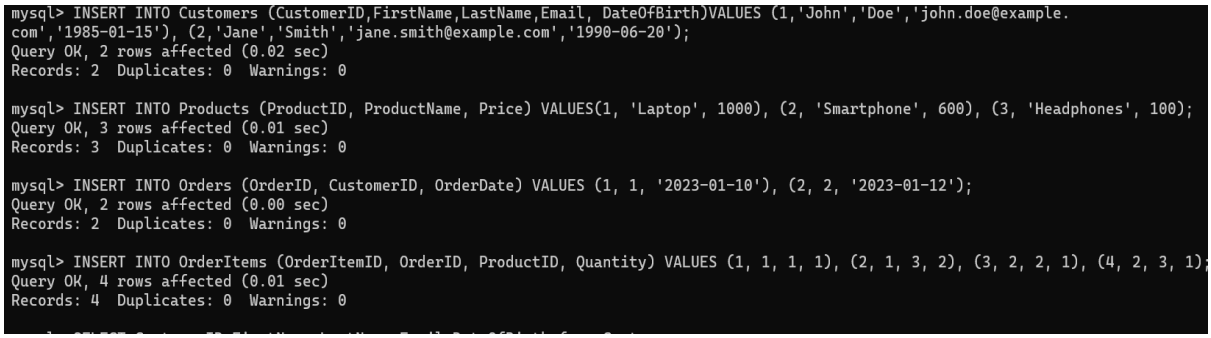
Sql queries outputs to GitHub

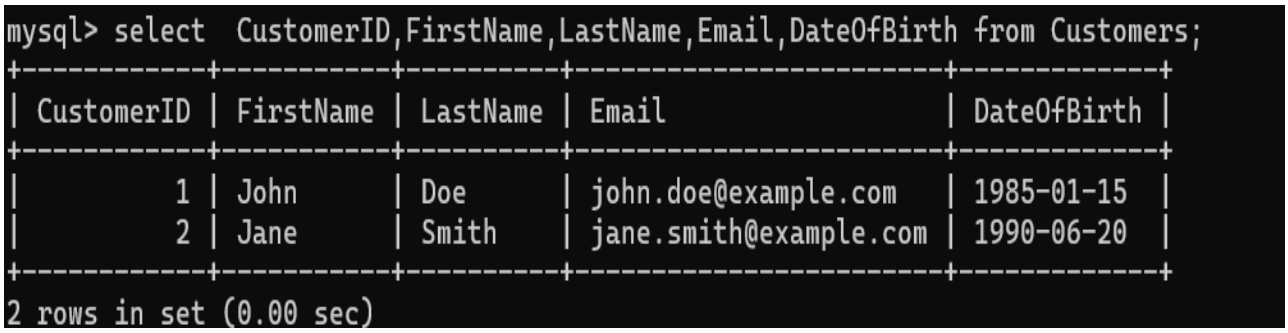
=>Insert the values into the database:



```
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> 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> INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES (1, 1, '2023-01-10'), (2, 2, '2023-01-12');
Query OK, 2 rows affected (0.00 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
```

1) List all customers.

   SELECT CustomerID, FirstName, LastName, Email, DateOfBirth FROM Customers;

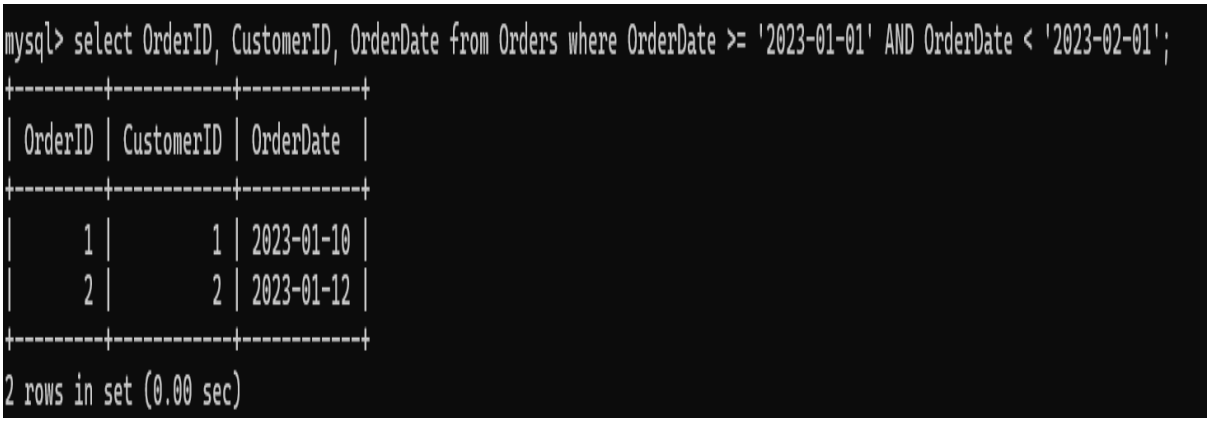


```
mysql> select  CustomerID,FirstName,LastName,Email,DateOfBirth from Customers;
+------------+-----------+----------+-----------------------+-------------+
| CustomerID | FirstName | LastName | Email                 | DateOfBirth |
+------------+-----------+----------+-----------------------+-------------+
|          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)
```

2) . Find all orders placed in January 2023.

Query:

select OrderID, CustomerID, OrderDate from Orders where OrderDate >= '2023-01-01' AND OrderDate < '2023-02-01';

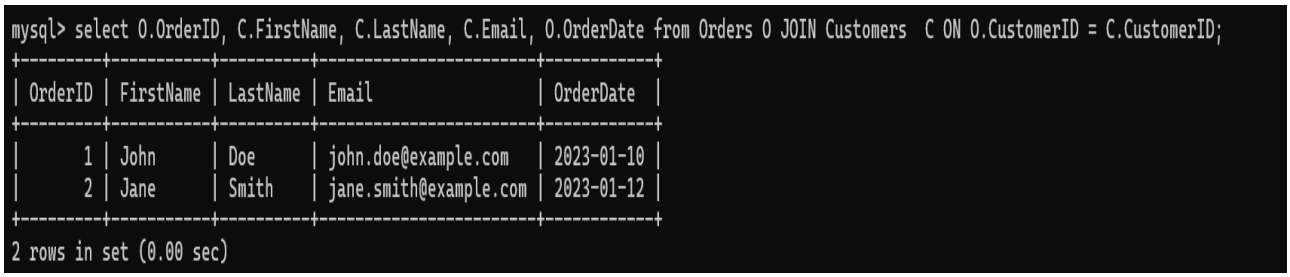


```
mysql> select OrderID, CustomerID, OrderDate from Orders where OrderDate >= '2023-01-01' AND OrderDate < '2023-02-01';
+---------+------------+-----------+
| OrderID | CustomerID | OrderDate |
+---------+------------+-----------+
|       1 |          1 | 2023-01-10 |
|       2 |          2 | 2023-01-12 |
+---------+------------+-----------+
2 rows in set (0.00 sec)
```

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

Query:

select O.OrderID, C.FirstName, C.LastName, C.Email, O.OrderDate from Orders O JOIN Customers C ON O.CustomerID = C.CustomerID;



```
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)
```

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

Query:

select P.ProductID, P.ProductName, P.Price, OI.Quantity from OrderItems OI join Products P ON OI.ProductID = P.ProductID WHERE OI.OrderID = 1;

```
mysql> select P.ProductID, P.ProductName, P.Price, OI.Quantity from OrderItems OI join Products  P ON OI.ProductID = P.ProductID WHERE OI.OrderID = 1;
+-----------+-------------+---------+----------+
| ProductID | ProductName | Price   | Quantity |
+-----------+-------------+---------+----------+
|         1 | Laptop      | 1000.00 |        1 |
|         3 | Headphones  |  100.00 |        2 |
+-----------+-------------+---------+----------+
2 rows in set (0.01 sec)
```

5) Calculate the total amount spent by each customer.

Query:

select C.CustomerID, C.FirstName, C.LastName, SUM(P.Price * OI.Quantity) AS Total Amount spent 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, C.FirstName, C.LastName;

```
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.Custo
merID join OrderItems OI on O.OrderID = OI.OrderID join Products P ON OI.ProductID = P.ProductID group by C.CustomerID, C.FirstName, C.LastName;
+------------+-----------+----------+-----------------+
| CustomerID | FirstName | LastName | TotalAmountSpent |
+------------+-----------+----------+-----------------+
|          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).

Query:

select p.ProductID, p.ProductName, SUM(oi.Quantity) AS TotalQuantity from OrderItems oi JOIN Products p ON oi.ProductID = p.ProductID GROUP BY p.ProductID, p.ProductName order by TotalQuantity DESC LIMIT 1;

```
mysql> select p.ProductID, p.ProductName, SUM(oi.Quantity) AS TotalQuantity from OrderItems oi JOIN Products p ON oi.ProductID = p.ProductID GROUP BY p.Prod
uctID, p.ProductName order by TotalQuantity desc limit 1;
+-----------+-------------+---------------+
| ProductID | ProductName | TotalQuantity |
+-----------+-------------+---------------+
|         3 | Headphones  |             3 |
+-----------+-------------+---------------+
1 row in set (0.01 sec)
```

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

Query:

select DATE_FORMAT(o.OrderDate, '%Y-%m') AS Month, COUNT(DISTINCT o.OrderID) AS TotalOrders, SUM(oi.Quantity * p.Price) AS TotalSalesAmount FROM Orders o join OrderItems oi ON o.OrderID = oi.OrderID JOIN Products p ON oi.ProductID = p.ProductID Where o.OrderDate >= '2023-01-01' AND o.OrderDate < '2024-01-01' group by DATE_FORMAT(o.OrderDate, '%Y-%m') order by Month;

```
mysql> select DATE_FORMAT(o.OrderDate, '%Y-%m') AS Month, COUNT(DISTINCT o.OrderID) AS TotalOrders, SUM(oi.Quantity * p.Price) AS TotalSalesAmount FROM Orde
rs o join OrderItems oi ON o.OrderID = oi.OrderID JOIN Products p ON oi.ProductID = p.ProductID Where o.OrderDate >= '2023-01-01' AND o.OrderDate < '2024-01
-01' group by DATE_FORMAT(o.OrderDate, '%Y-%m') order by Month;
+---------+------------+-----------------+
| Month   | TotalOrders | TotalSalesAmount |
+---------+------------+-----------------+
| 2023-01 |          2 |         1900.00 |
+---------+------------+-----------------+
1 row in set (0.01 sec)

mysql>
```

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

Query:

select c.CustomerID, c.FirstName, c.LastName, c.Email, SUM(oi.Quantity * p.Price) AS TotalSpent 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, c.FirstName, c.LastName, c.Email having SUM(oi.Quantity * p.Price) > 1000;