**SEU/IS/22/ICT/041**

A black and white emblem with a lion and an open book

AI-generated content may be incorrect.

**Subject code** : CIS11051

**Subject :** Practical of Database design

**Title:** Mysql Joins

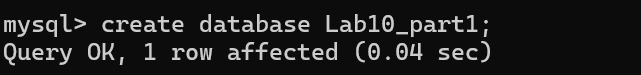
Department of Information Communication And Technology

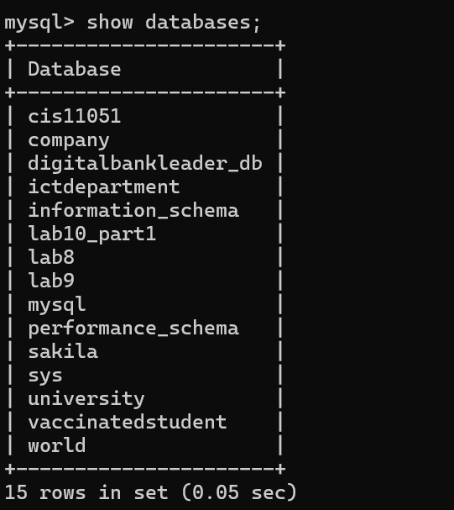
Faculty of Technology

Southeaster University of Sri Lanka

Exercise 1:

1. Create a database named Lab10\_Part1 and tables named Employees and Projects as shown below.

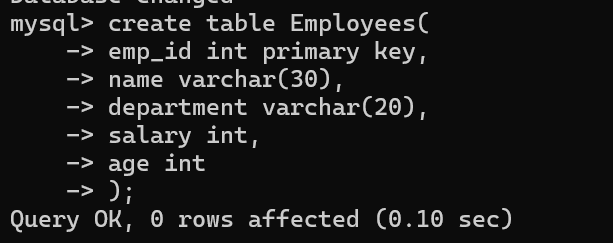
 Database Creation:

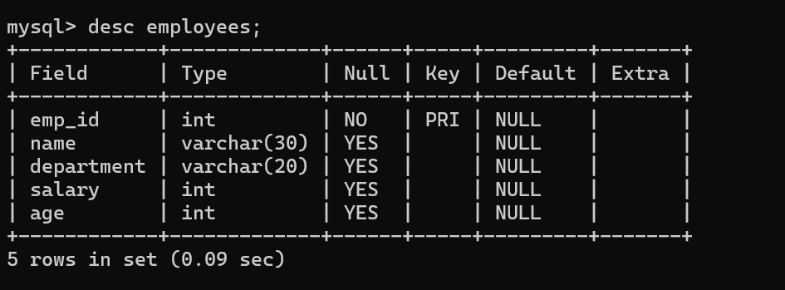


**Employees table:**

# Column Name Constraints

|  |  |
| --- | --- |
| *emp\_id name department salary age* | Primary key |
| Nullable |
| Nullable |
| Nullable |
| Nullable |

 Code:



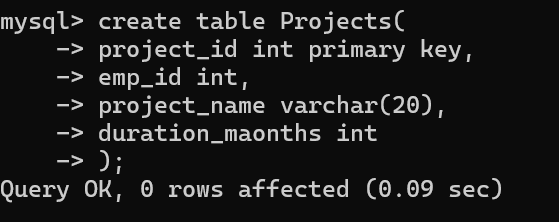
O

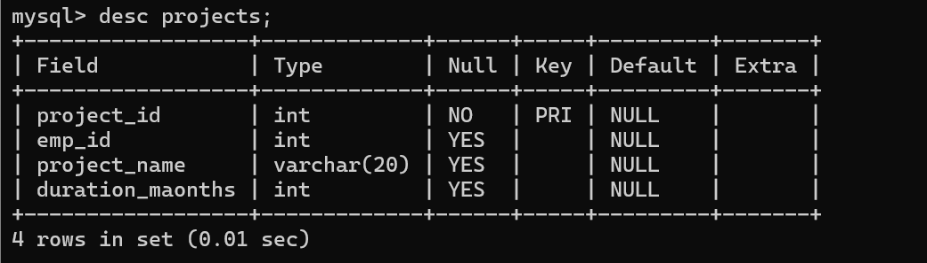
**Projects table:**

# Column Name Constraints

|  |  |
| --- | --- |
| *project\_id emp\_id project\_name duration\_months* | Primary key |
| Nullable |
| Nullable |
| Nullable |

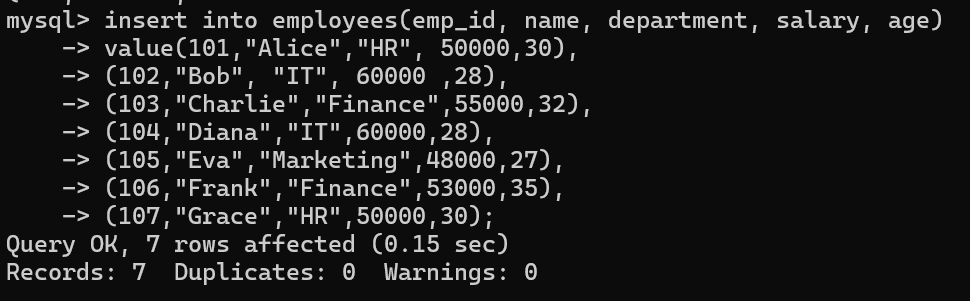
Code:





**Employees table:**

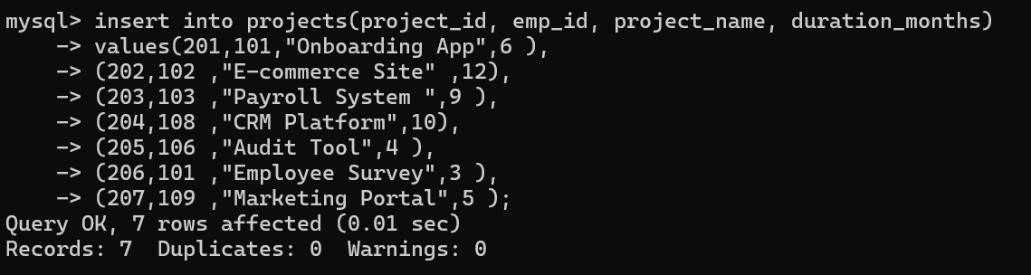
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **emp\_id** | **name** | **department** | **salary** | **age** |
| 101 | Alice | HR | 50000 | 30 |
| 102 | Bob | IT | 60000 | 28 |
| 103 | Charlie | Finance | 55000 | 32 |
| 104 | Diana | IT | 60000 | 28 |
| 105 | Eva | Marketing | 48000 | 27 |
| 106 | Frank | Finance | 53000 | 35 |
| 107 | Grace | HR | 50000 | 30 |

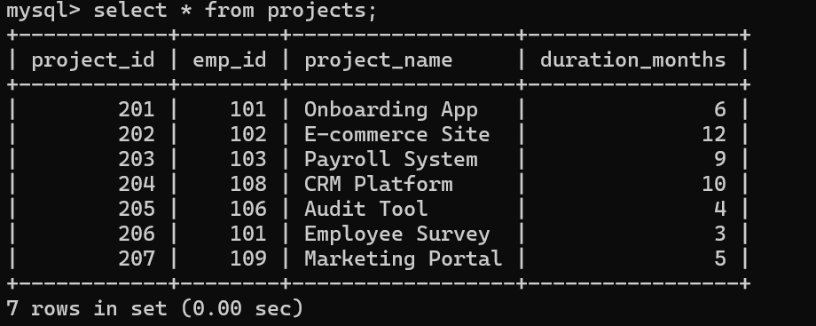
 Code:

**Projects table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **project\_id** | **emp\_id** | **project\_name** | **duration\_months** |
| 201 | 101 | Onboarding App | 6 |
| 202 | 102 | E-commerce Site | 12 |
| 203 | 103 | Payroll System | 9 |
| 204 | 108 | CRM Platform | 10 |
| 205 | 106 | Audit Tool | 4 |
| 206 | 101 | Employee Survey | 3 |
| 207 | 109 | Marketing Portal | 5 |

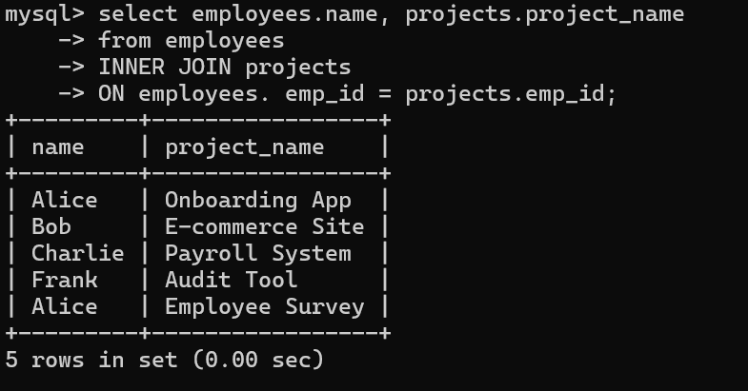
Code:



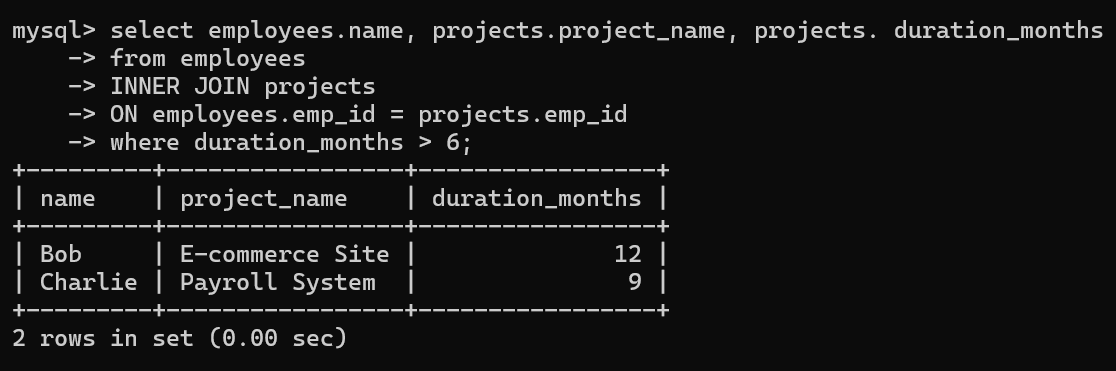


**Section A – INNER JOIN Focus**

1. List all employees’ names who are working on a project along with the project name.

Code:

1. Show all projects names with a duration greater than 6 months along with the corresponding employee names.

Code:

1. List employee names and project names where the salary is more than 55000.

A screenshot of a computer program

AI-generated content may be incorrect.Code:

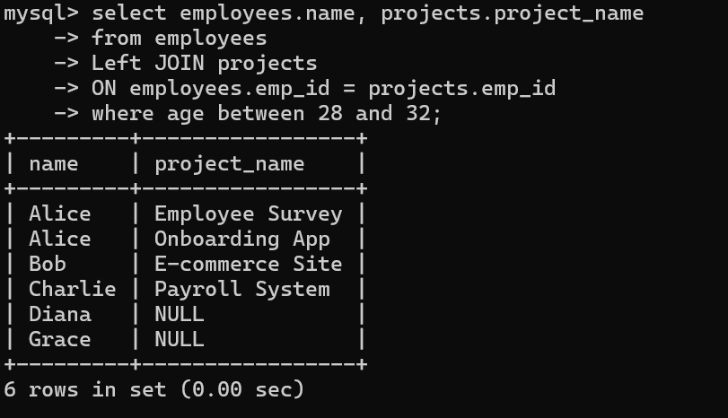
**Section B – LEFT JOIN Focus**

1. Display a list of all employee’s names and their assigned project names.

A screenshot of a computer program

AI-generated content may be incorrect.Code:

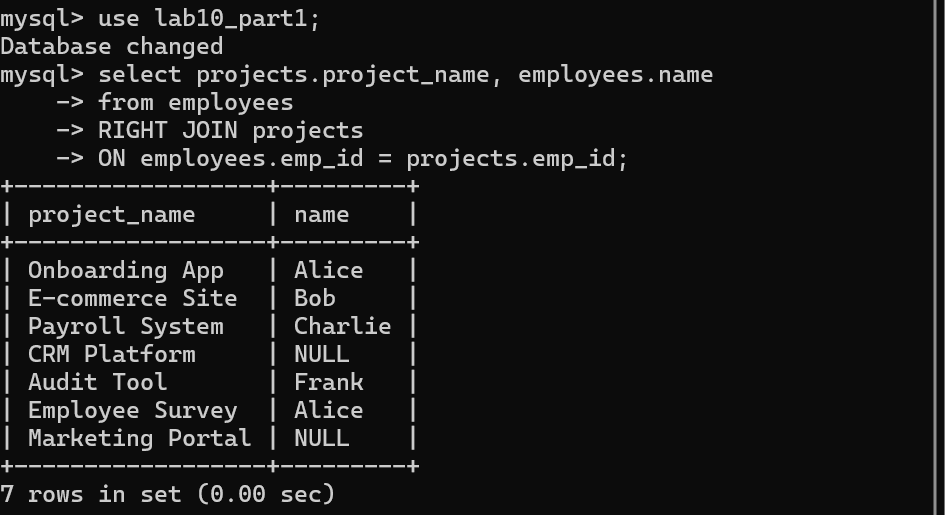
1. List the names of all employees and their project names if any, but only for employees whose age is between 28 and 32.

Code:

**Section C – RIGHT JOIN Focus**

1. Show all project names and their assigned employee names.

Code:



1. Display all project names and employee names (if assigned), where the project duration is more than 5 months OR the employee's salary is less than 55000.

A computer screen shot of a program

AI-generated content may be incorrect.Code:

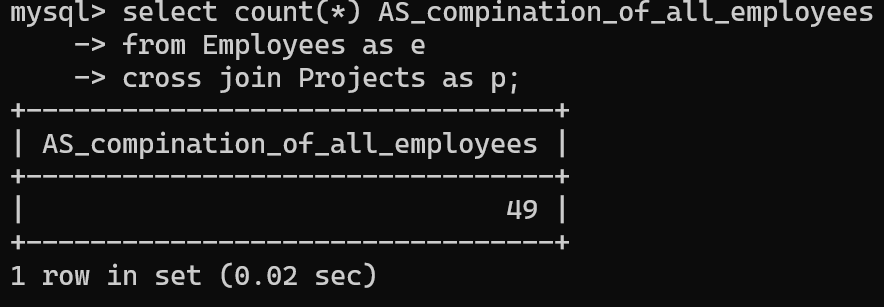
**Section D – CROSS JOIN Focus**

1. Create a cross combination of all employees with all project names (Cartesian product).

Code:

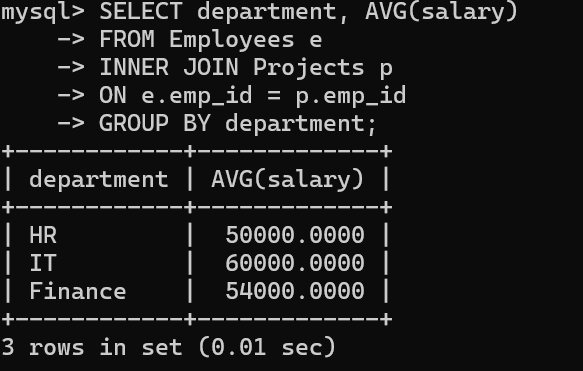


1. How many total combinations exist between employees and projects? Use aggregation.

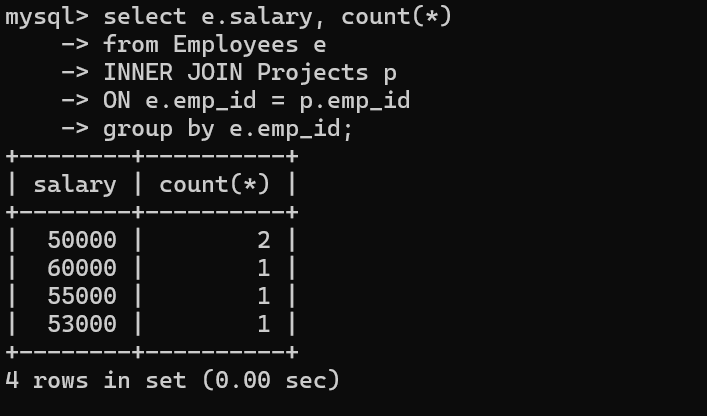


**Section E – Mixed Concepts**

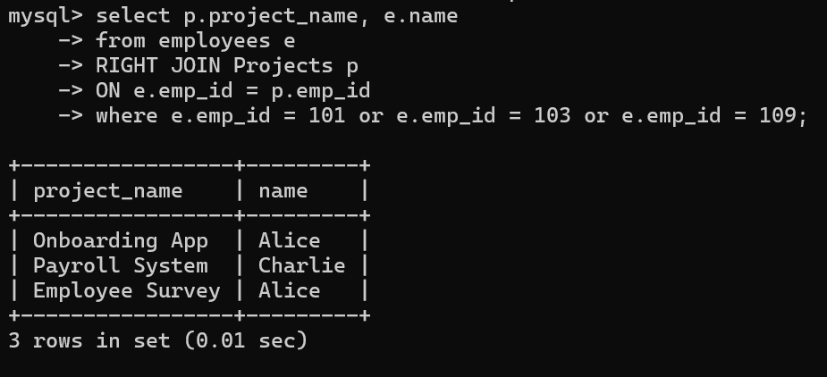
1. Find the average salary of employees in each department who have at least one project.

Code:

1. Find the number of employees with duplicate salaries and show their salary and count.

Code:

1. Show the project names and corresponding employee names (if available), but only for projects assigned to employee IDs in (101, 103, 109).

Code:

**Exercise 2 (ADVANCE):**

1. Create a **database** called **Lab10\_Part2** with two **tables**: **Customers and Orders**. Insert the given records using **appropriate data types**, establish a **foreign key relationship** between the tables, and use **INNER JOIN** in the queries to **retrieve the required results**.

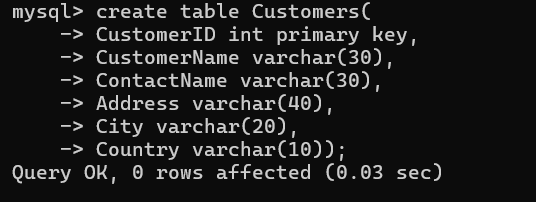
Database creation:



**Table: Customers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **ContactName** | **Address** | **City** | **Country** |
| 1 | Green Valley Farms | John Smith | 101 Maple Street | New York | USA |
| 2 | Oceanic Foods | Emma Johnson | 202 Ocean Drive | Los Angeles | USA |
| 3 | Sunny Cafe | Daniel Brown | 303 Sunset Blvd | Miami | USA |
| 4 | Tech World | Lisa Williams | 404 Tech Ave | San Jose | USA |
| 5 | Happy Pets | James Davis | 505 Pet Lane | Chicago | USA |
| 6 | The Book Nook | Sophia Miller | 606 Library Road | Boston | USA |
| 7 | Gourmet Market | Benjamin Wilson | 707 Gourmet Street | Houston | USA |
| 8 | Travel Experts | Olivia Martinez | 808 Travel Blvd | Seattle | USA |
| 9 | Bloom Florists | Alexander Garcia | 909 Rose Avenue | Denver | USA |
| 10 | Fast Fix Repairs | Charlotte Anderson | 1001 Fixer Street | Atlanta | USA |

**Table creation:**

****

**Table Describe:**A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a computer

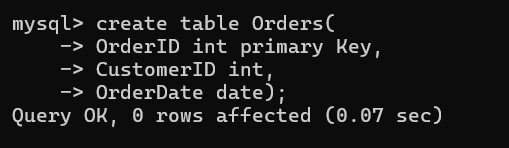
AI-generated content may be incorrect.Inserting Details:

A screenshot of a computer screen

AI-generated content may be incorrect.

**Table: Orders**

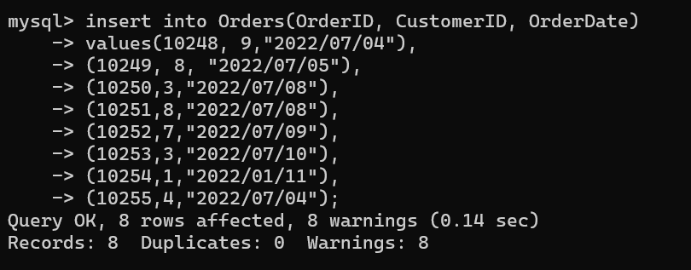
|  |  |  |
| --- | --- | --- |
| **OrderID** | **CustomerID** | **OrderDate** |
| 10248 | 9 | 7/4/2022 |
| 10249 | 8 | 7/5/2022 |
| 10250 | 3 | 7/8/2022 |
| 10251 | 8 | 7/8/2022 |
| 10252 | 7 | 7/9/2022 |
| 10253 | 3 | 7/10/2022 |
| 10254 | 1 | 7/11/2022 |
| 10255 | 4 | 7/4/2022 |

 Table creation:

A screen shot of a computer screen

AI-generated content may be incorrect.Describe Table:

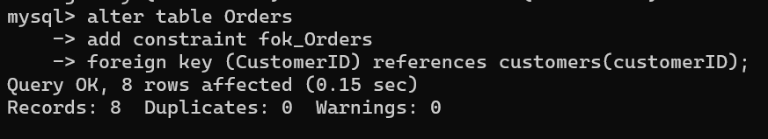
Inserting Details:

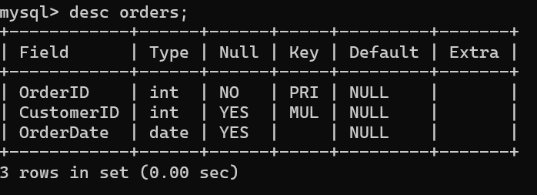
Code:

A screenshot of a computer screen

AI-generated content may be incorrect.

Foreign key :

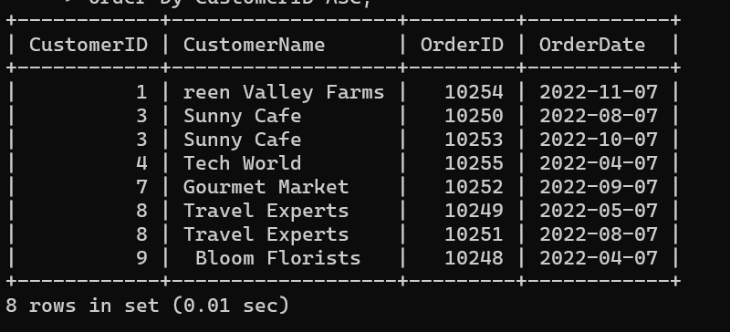
Out Output

****

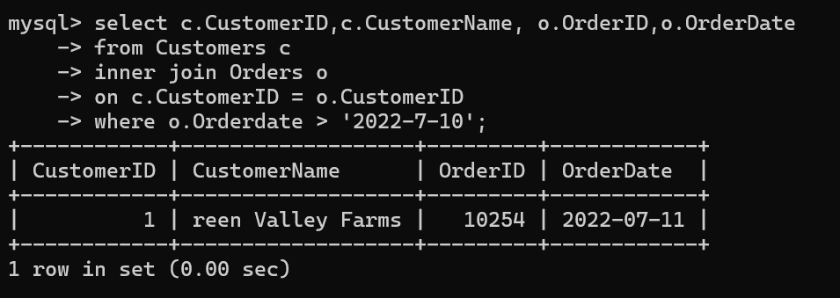
1. Show the details of customer ID, customer name from customer table, orderID and order date, only who made orders. And sort them by the customerID in ascending order.

A black screen with white text

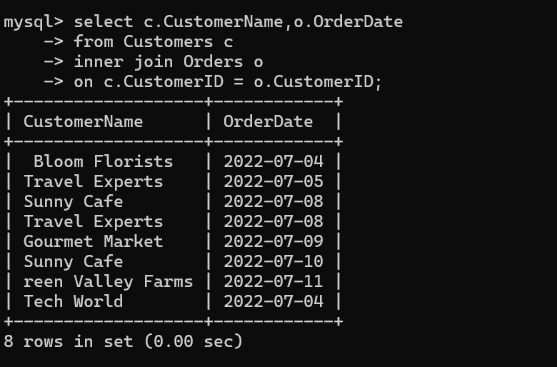
AI-generated content may be incorrect.Code:



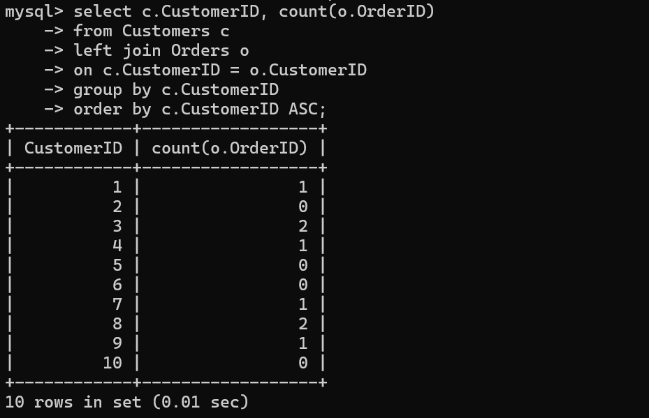
1. Display the customer details who made orders on after July10,2022.

Code:

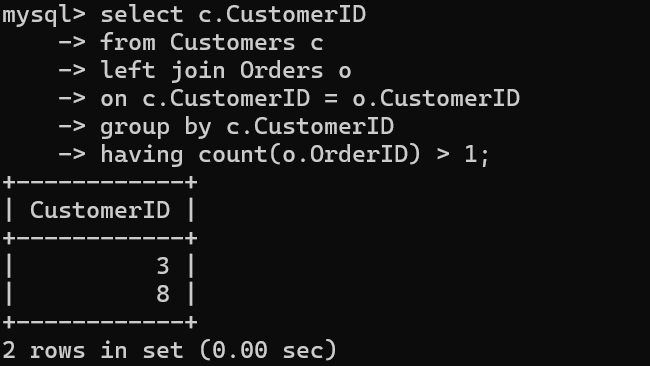
1. Display the details of customers who made orders and the date they made.

Code:

1. Count orders made by every customer and display the customer Id from customers and the number of orders made by them. Arrange the table in ascending order.

Code:

1. Display the customer Id from customer table who made more than one order. \

Code:

**Discussion:**

SQL JOINs play a crucial role in handling relational databases by combining related data from multiple tables. In a **customer order management system**, JOINs simplify retrieving structured data:

* **INNER JOIN**: Find customers who placed orders (it check both table related data).
* **LEFT JOIN**: Show all customers, even those without orders(only get left table related data).
* **RIGHT JOIN**: Display all orders, including those without customer details(Only get right table related data).
* **CROSS JOIN**: Generate all possible customer-product combinations for marketing analysis.

JOINs optimize queries, improve efficiency, and reduce redundant data processing, making them essential for real-world applications like e-commerce, finance, and customer management.