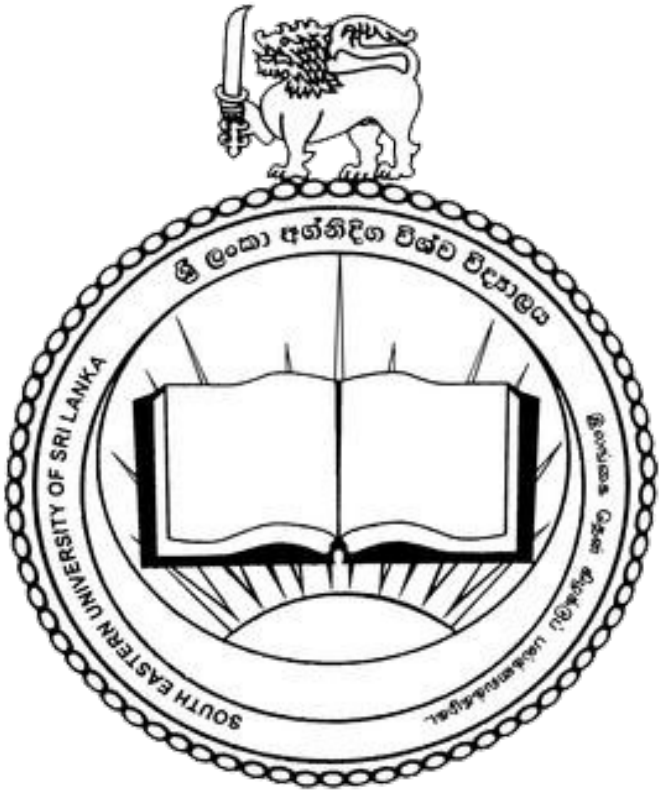
SEU/IS/22/ICT/041



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:

```
mysql> create database Lab10_part1;
Query OK, 1 row affected (0.04 sec)
```

```
mysql> show databases;
+----------------------+
| Database             |
+----------------------+
| cis11051             |
| company              |
| digitalbankleader_db |
| ictdepartment        |
| information_schema   |
| lab10_part1          |
| lab8                 |
| lab9                 |
| mysql                |
| performance_schema   |
| sakila               |
| sys                  |
| university           |
| vaccinatedstudent    |
| world                |
+----------------------+
15 rows in set (0.05 sec)
```

Employees table:

Column Name	*Constraints*
emp_id	Primary key
name	Nullable
department	Nullable
salary age	Nullable
	Nullable

Code:

```
mysql> create table Employees(
    -> emp_id int primary key,
    -> name varchar(30),
    -> department varchar(20),
    -> salary int,
    -> age int
    -> );
```

```
mysql> desc employees;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| emp_id     | int         | NO   | PRI | NULL    |       |
| name       | varchar(30) | YES  |     | NULL    |       |
| department | varchar(20) | YES  |     | NULL    |       |
| salary     | int         | YES  |     | NULL    |       |
| age        | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.09 sec)
```

O

Projects table:

Column Name	*Constraints*
project_id	Primary key
emp_id	Nullable
project_name	Nullable
duration_months	Nullable

Code:

```
mysql> create table Projects(
    -> project_id int primary key,
    -> emp_id int,
    -> project_name varchar(20),
    -> duration_maonths int
    -> );
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> desc projects;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| project_id       | int         | NO   | PRI | NULL    |       |
| emp_id           | int         | YES  |     | NULL    |       |
| project_name     | varchar(20) | YES  |     | NULL    |       |
| duration_maonths | int         | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

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:

```
mysql> insert into employees(emp_id, name, department, salary, age)
    -> value(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);
Query OK, 7 rows affected (0.15 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

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:

```
mysql> insert into projects(project_id, emp_id, project_name, duration_months)
    -> values(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 );
Query OK, 7 rows affected (0.01 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

```
mysql> select * from projects;
+------------+--------+------------------+-----------------+
| 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 |
+------------+--------+------------------+-----------------+
7 rows in set (0.00 sec)
```

Section A – INNER JOIN Focus

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

```
mysql> select employees.name, projects.project_name
    -> from employees
    -> INNER JOIN projects
    -> ON employees. emp_id = projects.emp_id;
+---------+-----------------+
| name    | project_name    |
+---------+-----------------+
| Alice   | Onboarding App  |
| Bob     | E-commerce Site |
| Charlie | Payroll System  |
| Frank   | Audit Tool      |
| Alice   | Employee Survey |
+---------+-----------------+
5 rows in set (0.00 sec)
```

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

```
mysql> select employees.name, projects.project_name, projects. duration_months
    -> from employees
    -> INNER JOIN projects
    -> ON employees.emp_id = projects.emp_id
    -> where duration_months > 6;
+---------+-----------------+-----------------+
| name    | project_name    | duration_months |
+---------+-----------------+-----------------+
| Bob     | E-commerce Site |              12 |
| Charlie | Payroll System  |               9 |
+---------+-----------------+-----------------+
2 rows in set (0.00 sec)
```

3. List employee names and project names where the salary is more than 55000.
   Code:

```
mysql> select employees.name, projects.project_name
    -> from employees
    -> INNER JOIN projects
    -> ON employees.emp_id = projects.emp_id
    -> where salary > 55000;
+------+-----------------+
| name | project_name    |
+------+-----------------+
| Bob  | E-commerce Site |
+------+-----------------+
1 row in set (0.00 sec)
```

Section B – LEFT JOIN Focus

1. Display a list of all employee's names and their assigned project names.
   Code:

```
mysql> select employees.name, projects.project_name
    -> from employees
    -> Left JOIN projects
    -> ON employees.emp_id = projects.emp_id;
+---------+-----------------+
| name    | project_name    |
+---------+-----------------+
| Alice   | Employee Survey |
| Alice   | Onboarding App  |
| Bob     | E-commerce Site |
| Charlie | Payroll System  |
| Diana   | NULL            |
| Eva     | NULL            |
| Frank   | Audit Tool      |
| Grace   | NULL            |
+---------+-----------------+
8 rows in set (0.00 sec)
```

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

```
mysql> select employees.name, projects.project_name
    -> from employees
    -> Left JOIN projects
    -> ON employees.emp_id = projects.emp_id
    -> where age between 28 and 32;
+---------+-----------------+
| name    | project_name    |
+---------+-----------------+
| Alice   | Employee Survey |
| Alice   | Onboarding App  |
| Bob     | E-commerce Site |
| Charlie | Payroll System  |
| Diana   | NULL            |
| Grace   | NULL            |
+---------+-----------------+
6 rows in set (0.00 sec)
```

Section C – RIGHT JOIN Focus

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

2. 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.
   Code:

```
mysql> select projects.project_name, employees.name
    -> from employees
    -> RIGHT JOIN projects
    -> ON employees.emp_id = projects.emp_id
    -> where duration_months > 5 and salary < 55000;
+-----------------+--------+
| project_name    | name   |
+-----------------+--------+
| Onboarding App | Alice |
+-----------------+--------+
1 row in set (0.00 sec)

mysql>
```

Section D – CROSS JOIN Focus

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

```
mysql> select e.emp_id, e.name, e.department, e.salary, e.age, p.project_name
    -> from Employees e
    -> cross join Projects p;
+--------+---------+------------+--------+------+------------------+
| emp_id | name    | department | salary | age  | project_name     |
+--------+---------+------------+--------+------+------------------+
|    107 | Grace   | HR         |  50000 |   30 | Onboarding App   |
|    106 | Frank   | Finance    |  53000 |   35 | Onboarding App   |
|    105 | Eva     | Marketing  |  48000 |   27 | Onboarding App   |
|    104 | Diana   | IT         |  60000 |   28 | Onboarding App   |
|    103 | Charlie | Finance    |  55000 |   32 | Onboarding App   |
|    102 | Bob     | IT         |  60000 |   28 | Onboarding App   |
|    101 | Alice   | HR         |  50000 |   30 | Onboarding App   |
|    107 | Grace   | HR         |  50000 |   30 | E-commerce Site  |
|    106 | Frank   | Finance    |  53000 |   35 | E-commerce Site  |
|    105 | Eva     | Marketing  |  48000 |   27 | E-commerce Site  |
|    104 | Diana   | IT         |  60000 |   28 | E-commerce Site  |
|    103 | Charlie | Finance    |  55000 |   32 | E-commerce Site  |
|    102 | Bob     | IT         |  60000 |   28 | E-commerce Site  |
|    101 | Alice   | HR         |  50000 |   30 | E-commerce Site  |
|    107 | Grace   | HR         |  50000 |   30 | Payroll System   |
|    106 | Frank   | Finance    |  53000 |   35 | Payroll System   |
|    105 | Eva     | Marketing  |  48000 |   27 | Payroll System   |
|    104 | Diana   | IT         |  60000 |   28 | Payroll System   |
|    103 | Charlie | Finance    |  55000 |   32 | Payroll System   |
|    102 | Bob     | IT         |  60000 |   28 | Payroll System   |
|    101 | Alice   | HR         |  50000 |   30 | Payroll System   |
|    107 | Grace   | HR         |  50000 |   30 | CRM Platform     |
|    106 | Frank   | Finance    |  53000 |   35 | CRM Platform     |
|    105 | Eva     | Marketing  |  48000 |   27 | CRM Platform     |
|    104 | Diana   | IT         |  60000 |   28 | CRM Platform     |
|    103 | Charlie | Finance    |  55000 |   32 | CRM Platform     |
|    102 | Bob     | IT         |  60000 |   28 | CRM Platform     |
|    101 | Alice   | HR         |  50000 |   30 | CRM Platform     |
|    107 | Grace   | HR         |  50000 |   30 | Audit Tool       |
|    106 | Frank   | Finance    |  53000 |   35 | Audit Tool       |
|    105 | Eva     | Marketing  |  48000 |   27 | Audit Tool       |
|    104 | Diana   | IT         |  60000 |   28 | Audit Tool       |
|    103 | Charlie | Finance    |  55000 |   32 | Audit Tool       |
|    102 | Bob     | IT         |  60000 |   28 | Audit Tool       |
|    101 | Alice   | HR         |  50000 |   30 | Audit Tool       |
|    107 | Grace   | HR         |  50000 |   30 | Employee Survey  |
|    106 | Frank   | Finance    |  53000 |   35 | Employee Survey  |
|    105 | Eva     | Marketing  |  48000 |   27 | Employee Survey  |
|    104 | Diana   | IT         |  60000 |   28 | Employee Survey  |
|    103 | Charlie | Finance    |  55000 |   32 | Employee Survey  |
|    102 | Bob     | IT         |  60000 |   28 | Employee Survey  |
|    101 | Alice   | HR         |  50000 |   30 | Employee Survey  |
|    107 | Grace   | HR         |  50000 |   30 | Marketing Portal |
|    106 | Frank   | Finance    |  53000 |   35 | Marketing Portal |
|    105 | Eva     | Marketing  |  48000 |   27 | Marketing Portal |
|    104 | Diana   | IT         |  60000 |   28 | Marketing Portal |
|    103 | Charlie | Finance    |  55000 |   32 | Marketing Portal |
|    102 | Bob     | IT         |  60000 |   28 | Marketing Portal |
|    101 | Alice   | HR         |  50000 |   30 | Marketing Portal |
+--------+---------+------------+--------+------+------------------+
49 rows in set (0.00 sec)
```

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

```
mysql> select count(*) AS_compination_of_all_employees
    -> from Employees as e
    -> cross join Projects as p;
+----------------------------------+
| AS_compination_of_all_employees |
+----------------------------------+
|                              49 |
+----------------------------------+
1 row in set (0.02 sec)
```

Section E – Mixed Concepts

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

```
mysql> SELECT department, AVG(salary)
    -> FROM Employees e
    -> INNER JOIN Projects p
    -> ON e.emp_id = p.emp_id
    -> GROUP BY department;
+------------+-------------+
| department | AVG(salary) |
+------------+-------------+
| HR         |  50000.0000 |
| IT         |  60000.0000 |
| Finance    |  54000.0000 |
+------------+-------------+
3 rows in set (0.01 sec)
```

2. Find the number of employees with duplicate salaries and show their salary and count.
   Code:

```
mysql> select e.salary, count(*)
    -> from Employees e
    -> INNER JOIN Projects p
    -> ON e.emp_id = p.emp_id
    -> group by e.emp_id;
+--------+----------+
| salary | count(*) |
+--------+----------+
|  50000 |        2 |
|  60000 |        1 |
|  55000 |        1 |
|  53000 |        1 |
+--------+----------+
4 rows in set (0.00 sec)
```

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

```
mysql> select p.project_name, e.name
    -> from employees e
    -> RIGHT JOIN Projects p
    -> ON e.emp_id = p.emp_id
    -> where e.emp_id = 101 or e.emp_id = 103 or e.emp_id = 109;

+-----------------+---------+
| project_name    | name    |
+-----------------+---------+
| Onboarding App  | Alice   |
| Payroll System  | Charlie |
| Employee Survey | Alice   |
+-----------------+---------+
3 rows in set (0.01 sec)
```

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:

```
mysql> create database Lab10_part2;
Query OK, 1 row affected (0.14 sec)
```

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:

```
mysql> create table Customers(
    -> CustomerID int primary key,
    -> CustomerName varchar(30),
    -> ContactName varchar(30),
    -> Address varchar(40),
    -> City varchar(20),
    -> Country varchar(10));
Query OK, 0 rows affected (0.03 sec)
```

Table Describe:

```
mysql> desc customers;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| CustomerID   | int         | NO   | PRI | NULL    |       |
| CustomerName | varchar(30) | YES  |     | NULL    |       |
| ContactName  | varchar(30) | YES  |     | NULL    |       |
| Address      | varchar(40) | YES  |     | NULL    |       |
| City         | varchar(20) | YES  |     | NULL    |       |
| Country      | varchar(10) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Inserting Details:

```
mysql> INSERT INTO Customers(CustomerID,CustomerName, ContactName, Address, City, Country)
    -> values(1,"reen 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");
Query OK, 10 rows affected (0.05 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> select * from customers;
+------------+------------------+--------------------+--------------------+-------------+---------+
| CustomerID | CustomerName     | ContactName        | Address            | City        | Country |
+------------+------------------+--------------------+--------------------+-------------+---------+
|          1 | reen 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     |
+------------+------------------+--------------------+--------------------+-------------+---------+
10 rows in set (0.00 sec)
```

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:

```
mysql> create table Orders(
    -> OrderID int primary Key,
    -> CustomerID int,
    -> OrderDate date);
Query OK, 0 rows affected (0.07 sec)
```

Describe Table:

```
mysql> desc orders;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| OrderID    | int  | NO   | PRI | NULL    |       |
| CustomerID | int  | YES  |     | NULL    |       |
| OrderDate  | date | YES  |     | NULL    |       |
+------------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Inserting Details:

Code:

```
mysql> insert into Orders(OrderID, CustomerID, OrderDate)
    -> values(10248, 9,"2022/07/04"),
    -> (10249, 8, "2022/07/05"),
    -> (10250,3,"2022/07/08"),
    -> (10251,8,"2022/07/08"),
    -> (10252,7,"2022/07/09"),
    -> (10253,3,"2022/07/10"),
    -> (10254,1,"2022/01/11"),
    -> (10255,4,"2022/07/04");
Query OK, 8 rows affected, 8 warnings (0.14 sec)
Records: 8  Duplicates: 0  Warnings: 8
```

```
mysql> select * from Orders;
+---------+------------+------------+
| OrderID | CustomerID | OrderDate  |
+---------+------------+------------+
|   10248 |          9 | 2022-07-04 |
|   10249 |          8 | 2022-07-05 |
|   10250 |          3 | 2022-07-08 |
|   10251 |          8 | 2022-07-08 |
|   10252 |          7 | 2022-07-09 |
|   10253 |          3 | 2022-07-10 |
|   10254 |          1 | 2022-01-11 |
|   10255 |          4 | 2022-07-04 |
+---------+------------+------------+
8 rows in set (0.00 sec)
```

Foreign key :

Out

```
mysql> alter table Orders
    -> add constraint fok_Orders
    -> foreign key (CustomerID) references customers(customerID);
Query OK, 8 rows affected (0.15 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

```
mysql> desc orders;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| OrderID    | int  | NO   | PRI | NULL    |       |
| CustomerID | int  | YES  | MUL | NULL    |       |
| OrderDate  | date | YES  |     | NULL    |       |
+------------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

2. 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.

Code:

```
mysql> select c.CustomerID, c.CustomerName, o.OrderID, o.OrderDate
    -> from Customers c
    -> INNER JOIN Orders o
    -> ON c.CustomerID = o.CustomerID
    -> order by CustomerID ASC;
```

```
+------------+------------------+---------+------------+
| CustomerID | CustomerName     | OrderID | OrderDate  |
+------------+------------------+---------+------------+
|          1 | reen Valley Farms |   10254 | 2022-11-07 |
|          3 | Sunny Cafe       |   10250 | 2022-08-07 |
|          3 | Sunny Cafe       |   10253 | 2022-10-07 |
|          4 | Tech World       |   10255 | 2022-04-07 |
|          7 | Gourmet Market   |   10252 | 2022-09-07 |
|          8 | Travel Experts   |   10249 | 2022-05-07 |
|          8 | Travel Experts   |   10251 | 2022-08-07 |
|          9 |  Bloom Florists  |   10248 | 2022-04-07 |
+------------+------------------+---------+------------+
8 rows in set (0.01 sec)
```

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

Code:

```
mysql> select c.CustomerID,c.CustomerName, o.OrderID,o.OrderDate
    -> from Customers c
    -> inner join Orders o
    -> on c.CustomerID = o.CustomerID
    -> where o.Orderdate > '2022-7-10';
+------------+------------------+---------+------------+
| CustomerID | CustomerName     | OrderID | OrderDate  |
+------------+------------------+---------+------------+
|          1 | reen Valley Farms |   10254 | 2022-07-11 |
+------------+------------------+---------+------------+
1 row in set (0.00 sec)
```

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

Code:

```
mysql> select c.CustomerName,o.OrderDate
    -> from Customers c
    -> inner join Orders o
    -> on c.CustomerID = o.CustomerID;
+------------------+------------+
| CustomerName     | OrderDate  |
+------------------+------------+
|  Bloom Florists  | 2022-07-04 |
| Travel Experts   | 2022-07-05 |
| Sunny Cafe       | 2022-07-08 |
| Travel Experts   | 2022-07-08 |
| Gourmet Market   | 2022-07-09 |
| Sunny Cafe       | 2022-07-10 |
| reen Valley Farms | 2022-07-11 |
| Tech World       | 2022-07-04 |
+------------------+------------+
8 rows in set (0.00 sec)
```

5. 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:

```
mysql> select c.CustomerID, count(o.OrderID)
    -> from Customers c
    -> left join Orders o
    -> on c.CustomerID = o.CustomerID
    -> group by c.CustomerID
    -> order by c.CustomerID ASC;
+------------+------------------+
| CustomerID | count(o.OrderID) |
+------------+------------------+
|          1 |                1 |
|          2 |                0 |
|          3 |                2 |
|          4 |                1 |
|          5 |                0 |
|          6 |                0 |
|          7 |                1 |
|          8 |                2 |
|          9 |                1 |
|         10 |                0 |
+------------+------------------+
10 rows in set (0.01 sec)
```

6. Display the customer Id from customer table who made more than one order. \
   Code:

```
mysql> select c.CustomerID
    -> from Customers c
    -> left join Orders o
    -> on c.CustomerID = o.CustomerID
    -> group by c.CustomerID
    -> having count(o.OrderID) > 1;
+-------------+
| CustomerID |
+-------------+
|           3 |
|           8 |
+-------------+
2 rows in set (0.00 sec)
```

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.