# GEN-AI ASSIGNMENT-4

1. **Write a query to give inner join,left outer join,right outer join and full outer join**

CREATE TABLES EMPLOYEE,DEPARTMENT

**EMPLOYEE TABLE**

```
CREATE TABLE departments (
   department_id INT PRIMARY KEY,
   department_name VARCHAR(50)
);
```

**DEPARTMENT TABLE**

```
CREATE TABLE employees (
   employee_id INT PRIMARY KEY,
   first_name VARCHAR(50),
   last_name VARCHAR(50),
   department_id INT,
   FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

**INSERTING DATA INTO EMPLOYEE TABLE**

```
INSERT INTO departments (department_id, department_name) VALUES
(1, 'HR'),
(2, 'Engineering'),
(3, 'Marketing');
```

**INSERTING DATA INTO DEPARTMENT TABLE**

```
INSERT INTO employees (employee_id, first_name, last_name, department_id)
VALUES
(1, 'John', 'Doe', 1),
(2, 'Jane', 'Smith', 2),
(3, 'Mike', 'Johnson', 3),
(4, 'Emily', 'Davis', 2),
(5, 'Anna', 'Brown', NULL);
```

# PERFORMING DIFFERENT JOINTS

**INNER JOIN:** An inner join returns rows when there is a match in both tables

**SQL QUERY:**

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM employees e
INNER JOIN departments d ON e.department_id = d.department_id;
```

**OUTPUT:**

| employee_id | first_name | last_name | department_name |
|---|---|---|---|
| 1 | John | Doe | HR |
| 2 | Jane | Smith | Engineering |
| 3 | Mike | Johnson | Marketing |
| 4 | Emily | Davis | Engineering |

**LEFT OUTER JOIN:** A left outer join returns all rows from the left table (employees), and the matched rows from the right table (departments). If no match, NULL values are returned for columns of the right table.

**SQL QUERY:**

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM employees e
LEFT JOIN departments d ON e.department_id = d.department_id;
```

**OUTPUT:**

| employee_id | first_name | last_name | department_name |
|---|---|---|---|
| 1 | John | Doe | HR |
| 2 | Jane | Smith | Engineering |
| 3 | Mike | Johnson | Marketing |
| 4 | Emily | Davis | Engineering |
| 5 | Anna | Brown | NULL |

**RIGHT OUTER JOIN:** A right outer join returns all rows from the right table (departments), and the matched rows from the left table (employees). If no match, NULL values are returned for columns of the left table.

**SQL QUERY:**

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM employees e
RIGHT JOIN departments d ON e.department_id = d.department_id;
```

**OUTPUT:**

| employee_id | first_name | last_name | department_name |
|---|---|---|---|
| 1 | John | Doe | HR |
| 2 | Jane | Smith | Engineering |
| 4 | Emily | Davis | Engineering |
| 3 | Mike | Johnson | Marketing |

**FULL OUTER JOIN**: A full outer join returns all rows when there is a match in either left (employees) or right (departments) table records. If there is no match, the result is NULL from the side where there is no match.

**SQL QUERY:**

```
SELECT e.employee_id, e.first_name, e.last_name, d.department_name
FROM employees e
FULL OUTER JOIN departments d ON e.department_id = d.department_id;
```

**OUTPUT:**

| employee_id | first_name | last_name | department_name |
|---|---|---|---|
| 1 | John | Doe | HR |
| 2 | Jane | Smith | Engineering |
| 3 | Mike | Johnson | Marketing |
| 4 | Emily | Davis | Engineering |
| 5 | Anna | Brown | NULL |
| NULL | NULL | NULL | Sales |