Experiment 2

Name: Sumit Kumar **UID: 23BCS11219**
Branch: B.E.CSE **Section/Group: 23BCS-KRG-3B**
Semester: 5th **Date of Performance:28-07-25**
Subject Name: ADBMS **Subject Code:23CSP-333**

1. Aim:

a) You are a Database Engineer at TalentTree Inc., an enterprise HR analytics platform that stores employee data, including their reporting relationships. The company maintains a centralized Employee relation that holds:
Each employee's ID, name, department, and manager ID (who is also an employee in the same table).
Your task is to generate a report that maps employees to their respective managers, showing:
The employee's name and department
Their manager's name and department (if applicable)
This will help the HR department visualize the internal reporting hierarchy.

However, not all ID-YEAR combinations in the Queries table are present in the Year_tbl. If an NPV is missing for a requested combination, assume it to be 0 to maintain a consistent financial report.

2. Objective:

- To understand how to use JOINS in SQL.
- To understand the basic SQL Queries.
- To generate hierarchical reports from self-referencing tables.

3. DBMS script and output:

```
CREATE TABLE Employee (
   EmpID INT,
   Ename VARCHAR(100),
   Department VARCHAR(100),
   ManagerID INT
);

INSERT INTO Employee VALUES
```

```
(1, 'Anjali', 'Marketing', NULL),
(2, 'Rohan', 'Sales', 1),
(3, 'Meera', 'Tech', 1),
(4, 'Arjun', 'Sales', 2),
(5, 'Priya', 'Tech', 3),
(6, 'Neha', 'Marketing', 1);

SELECT
   E1.Ename AS [Employee Name],
   E1.Department AS [Employee Department],
   E2.Ename AS [Manager Name],
   E2.Department AS [Manager Department]
FROM
   Employee AS E1
LEFT JOIN
   Employee AS E2
ON
   E1.ManagerID = E2.EmpID;
```

4. Output:

Employee Name	Employee Department	Manager Name	Manager Department
Anjali	Marketing		
Rohan	Sales	Anjali	Marketing
Meera	Tech	Anjali	Marketing
Arjun	Sales	Rohan	Sales
Priya	Tech	Meera	Tech
Neha	Marketing	Anjali	Marketing

5. Learning outcomes:

- You will be able to write basic SQL queries.
- You will learn to perform JOINS in SQL.
- You will understand how to implement foreign k