DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Experiment- 2

Student Name: RAVI **UID:23BCS10340**
Branch: CSE **Section/Group: KRG-3A**
Semester: 5th **Date of Performance: 25 JUL**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1) Problem Statement :

Organizational Hierarchy Explorer

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.

TOOLS USED : Microsoft SQL Server

SQL CODE :

```
CREATE TABLE Employee (
   EmpID INT PRIMARY KEY,
   EmpName VARCHAR(50) NOT NULL,
   Department VARCHAR(50) NOT NULL,
   ManagerID INT NULL
);

ALTER TABLE Employee
ADD CONSTRAINT FK_Manager FOREIGN KEY (ManagerID) REFERENCES
Employee(EmpID);

INSERT INTO Employee (EmpID, EmpName, Department, ManagerID)
```

```
VALUES
(1, 'Ravi', 'HR', NULL),
(2, 'Sneha', 'Finance', 1),
(3, 'Karan', 'IT', 1),
(4, 'Meena', 'Finance', 2),
(5, 'Arjun', 'IT', 3),
(6, 'Pooja', 'HR', 1);

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

OUTPUT:

Employee Name	Employee Dept	Manager Name	Manager Dept
Ravi	HR	NULL	NULL
Sneha	Finance	Ravi	HR
Karan	IT	Ravi	HR
Meena	Finance	Sneha	Finance
Arjun	IT	Karan	IT
Pooja	HR	Ravi	HR

2) Problem Statement :

Financial Forecast Matching with Fallback Strategy

You are a Data Engineer at **FinSight Corp**, a company that models Net Present Value (NPV) projections for investment decisions. Your system maintains two key datasets:

1. **Year_tbl:** Actual recorded NPV's of various financial instruments over different years:

   ID: Unique Financial instrument identifier.

   YEAR: Year of record

   NPV: Net Present Value in that year

2. **Queries_tbl:** A list of instrument-year pairs for which stakeholders are requesting NPV values:

   ID: Financial instrument identifier

   YEAR: Year of interest.

Find the NPV of each query from the Queries table. Return the output order by ID and Year in the sorted form.

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.

TOOLS USED : Microsoft SQL Server

SQL CODE:

```
CREATE TABLE YEAR_TABLE(

ID INT,

YEAR INT,

NPV INT

);
```

```
INSERT INTO YEAR_TABLE(ID,YEAR,NPV)
VALUES
(1,2018,100),
(7,2020,30),
(13,2019,40),
(1,2019,113),
(2,2008,121),
(3,2009,12),
(11,2020,99),
(7,2019,0);

CREATE TABLE QUERIES_TABLE(
ID INT,
YEAR INT
);
INSERT INTO QUERIES_TABLE( ID,YEAR)
VALUES
(1,2019),
(2,2008),
(3,2009),
(7,2018),
(7,2019),
```

(7,2020),

(13,2019);

```
SELECT Q.ID,Q.YEAR,ISNULL(Y.NPV,0) AS[NPV]
FROM QUERIES_TABLE AS Q
LEFT OUTER JOIN
YEAR_TABLE AS Y
ON
Q.ID = Y.ID
AND
Y.YEAR = Q.YEAR;
```

OUTPUT :

ID	YEAR	NPV
1	2019	113
2	2008	121
3	2009	12
7	2018	0
7	2019	0
7	2020	30
13	2019	40