



## **Experiment – 2**

**Student Name: Tarak Khurana**

**Branch: BE-CSE**

**Semester: 5th**

**Subject Name: ADBMS**

**UID: 23BCS12145**

**Section/Group: KRG\_1A**

**Date of Performance: 24-07-25**

**Subject Code: 23CSP-333**

### **1. Aim:**

#### **Q1) 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.

#### **Q2) NPV Lookup Using Left Join**

You have two tables:

1. **Year tbl:** contains actual NPV (Net Present Value) for different years and IDs.
2. **Queries:** contains (ID, YEAR) pairs for which you want to find the NPV.

**Goal:** Return the queried (ID, YEAR) pairs along with the corresponding NPV value.

If NPV doesn't exist for that (ID, YEAR), show 0.

### **2. Requirements (Hardware/Software):**

MySQL, PostgreSQL, Oracle, or SQL Server

### **3. DBMS Script & Output:**

[MEDIUM LEVEL]

--Organizational Hierarchy Explorer



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
CREATE TABLE EMPLOYEE (  
    EmpID int,
```

```
    Ename varchar(15),  
    Department varchar(10),  
    ManagerID int
```

```
);
```

```
INSERT INTO EMPLOYEE VALUES (1, 'Alice', 'HR',NULL);  
INSERT INTO EMPLOYEE VALUES (2, 'Bob', 'Finance',1);  
INSERT INTO EMPLOYEE VALUES (3, 'Charlie', 'IT',1);  
INSERT INTO EMPLOYEE VALUES (4, 'David', 'Finance',2);  
INSERT INTO EMPLOYEE VALUES (5, 'Eve', 'IT',3);  
INSERT INTO EMPLOYEE VALUES (6, 'Frank', 'HR',1);
```

```
--SELECT * FROM EMPLOYEE;
```

```
SELECT E.Ename AS [EMPLOYEE NAME],E2.Ename AS [MANAGER  
    NAME],  
E.Department AS [EMPLOYEE DEPARTMENT],E2.DEPARTMENT AS  
    [MANAGER DEPARTMENT]  
FROM  
EMPLOYEE AS E  
LEFT JOIN  
EMPLOYEE AS E2  
ON  
E.ManagerID=E2.EmpID
```

[HARD LEVEL]

```
CREATE TABLE Year_tbl(  
    ID int,  
    YEAR int,
```

```
NPV int,  
);  
TRUNCATE TABLE Year_tbl;  
INSERT INTO Year_tbl VALUES(1,2018,100);  
INSERT INTO Year_tbl VALUES(7,2020,30);  
INSERT INTO Year_tbl VALUES(13,2019,40);  
INSERT INTO Year_tbl VALUES(1,2019,113);  
INSERT INTO Year_tbl VALUES(2,2008,121);  
INSERT INTO Year_tbl VALUES(3,2009,12);  
INSERT INTO Year_tbl VALUES(11,2020,99);  
INSERT INTO Year_tbl VALUES(7,2019,0);  
  
--SELECT * FROM Year_tbl;  
  
CREATE TABLE Queries_tbl(  
    ID int,  
    YEAR int  
);  
INSERT INTO Queries_tbl VALUES(1,2019);  
INSERT INTO Queries_tbl VALUES(2,2008);  
INSERT INTO Queries_tbl VALUES(3,2009);  
INSERT INTO Queries_tbl VALUES(7,2018);  
INSERT INTO Queries_tbl VALUES(7,2019);  
INSERT INTO Queries_tbl VALUES(7,2020);  
INSERT INTO Queries_tbl VALUES(13,2019);  
--SELECT * FROM Queries_tbl;  
  
SELECT Q.ID,Q.YEAR,ISNULL(Y.NPV,0)AS NPV  
FROM Queries_tbl AS Q  
LEFT OUTER JOIN  
Year_tbl AS Y
```

ON

Y.ID=Q.ID

and

Y.YEAR=Q.YEAR

#### 4. Output:

[MEDIUM LEVEL]

| EMPLOYEE NAME | EMPLOYEE DEPT | MANAGER NAME | MANAGER DEPT |
|---------------|---------------|--------------|--------------|
| Frank         | HR            | Alice        | HR           |
| Charlie       | IT            | Alice        | HR           |
| Bob           | Finance       | Alice        | HR           |
| David         | Finance       | Bob          | Finance      |
| Eve           | IT            | Charlie      | IT           |
| Alice         | HR            | -            | -            |

[HARD LEVEL]

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

#### 5. LEARNING OUTCOMES:

- Gained practical experience in table creation and record manipulation in databases.
- Learned how to join tables and extract relevant values efficiently.
- Understood implementation of OUTER JOINS to integrate and analyze related data.