



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

EXPERIMENT - 2

Student Name: Shivansh

UID: 23BCS13205

Branch: BE-CSE

Section/Group: KRG2-A

Semester: 5th

Subject Name: ADBMS

Subject Code: 23CSP-333

1. **AIM:**

Ques 1 :- 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.

| EmpID | Ename | Department | ManagerID |
|-------|---------|------------|-----------|
| 1 | Alice | HR | NULL |
| 2 | Bob | Finance | 1 |
| 3 | Charlie | IT | 1 |
| 4 | David | Finance | 2 |
| 5 | Eve | IT | 3 |
| 6 | Frank | HR | 1 |



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

2. TOOLS USED:- MS SSMS & Microsoft SQL Server

3. 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, 'Alice', 'HR', NULL),  
(2, 'Bob', 'Finance', 1),  
(3, 'Charlie', 'IT', 1),  
(4, 'David', 'Finance', 2),  
(5, 'Eve', 'IT', 3),  
(6, 'Frank', 'HR', 1);  
  
SELECT  
    E.EmpName AS [EmployeeName],  
    E.Department AS [EmployeeDept],  
    M.EmpName AS [Manager Name],  
    M.Department AS [ManagerDept]  
FROM Employee AS E  
JOIN Employee AS M
```

ON E.ManagerId = M.EmpID;

4. OUTPUT:

| | EmployeeName | EmployeeDept | Manager Name | ManagerDept |
|---|--------------|--------------|--------------|-------------|
| 1 | Bob | Finance | Alice | HR |
| 2 | Charlie | IT | Alice | HR |
| 3 | David | Finance | Bob | Finance |
| 4 | Eve | IT | Charlie | IT |
| 5 | Frank | HR | Alice | HR |

5. Ques 2: -Financial Forecast Matching with Fallback Strategy (hard)

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.

| ID | YEAR | NPV | ID | YEAR |
|----|------|-----|----|------|
| 1 | 2018 | 100 | 1 | 2019 |
| 7 | 2020 | 30 | 2 | 2008 |
| 13 | 2019 | 40 | 3 | 2009 |
| 1 | 2019 | 113 | 7 | 2018 |
| 2 | 2008 | 121 | 7 | 2019 |
| 3 | 2009 | 12 | 7 | 2020 |
| 11 | 2020 | 99 | 13 | 2019 |
| 7 | 2019 | 0 | | |

Year Table

Queries Table



6. SQL CODE:-

```
CREATE TABLE YEARS_TBL(  
    ID INT,  
    YEAR INT,  
    NPV INT  
)  
INSERT INTO YEARS_TBL(ID, YEAR, NPV)  
VALUES  
(1,2018,100),  
(7,2020,30),  
(13,2019,40),  
(1,2019,113),  
(2,2008,121),  
(3,2002,12),  
(11,2020,99),  
(7,2019,0);
```

```
CREATE TABLE QUERIES_TBL(  
    ID INT,  
    YEAR INT  
);  
INSERT INTO QUERIES_TBL(ID, YEAR)  
VALUES  
(1,2019),  
(2,2008),  
(3,2009),  
(7,2018),  
(7,2019),  
(7,2020),  
(13,2019);
```

```
SELECT Q.*,ISNULL(Y.NPV,0) AS [NPV]  
FROM  
YEARS_TBL AS Y
```

RIGHT OUTER JOIN
QUERIES_TBL AS Q
ON
Y.ID = Q.ID
AND
Y.YEAR = Q.YEAR

7. OUTPUT

| | ID | YEAR | NPV |
|---|----|------|-----|
| 1 | 1 | 2019 | 113 |
| 2 | 2 | 2008 | 121 |
| 3 | 3 | 2009 | 0 |
| 4 | 7 | 2018 | 0 |
| 5 | 7 | 2019 | 0 |
| 6 | 7 | 2020 | 30 |
| 7 | 13 | 2019 | 40 |