



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment 2

**Student Name:** Sohit Kumar

**Branch:** CSE

**Semester:** 5th

**Subject Name:** ADBMS

**UID:** 23BCS12492

**Section/Group:** KRG-3A

**Date of Performance:** 24 JUL

**Subject Code:** 23CSP-333

### 1. Problem Statement & SQL Code:

#### 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.

### Solution:

```
CREATE TABLE Employee (  
    EmpID INT PRIMARY KEY,  
    Ename VARCHAR(100),  
    Department VARCHAR(50),  
    ManagerId INT  
    FOREIGN KEY (ManagerId) REFERENCES Employee(EmpID)  
);
```

```
INSERT INTO Employee (EmpID, Ename, Department, ManagerId) VALUES
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

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

```
SELECT E1.Ename AS [EmployeeName], E2.Ename AS [ManagerName],  
E1.Department AS [Employee_Dept], E2.DEPT AS [ManagerDept]  
FROM  
Employee AS E1  
LEFT OUTER JOIN  
Employee AS E2  
ON  
E1.ManagerId = E2.EmpID
```

## OUTPUT:

	EMPLOYEEENAME	EMPLOYEE_DEPT	MANAGERNAME	MANAGERDEPT
1	Bob	Finance	Alice	HR
2	Charlie	IT	Alice	HR
3	David	Finance	Bob	Finance
4	Frank	HR	Bob	Finance
5	Eve	IT	Charlie	IT
6	Alice	HR	(null)	(null)

## Q2) 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:

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

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.

### Solution:

```
CREATE TABLE Year_tbl (  
    ID INT,  
    YEAR INT,  
    NPV INT  
);
```

```
CREATE TABLE Queries (  
    ID INT,  
    YEAR INT  
);
```

```
INSERT INTO Year_tbl (ID, YEAR, NPV) VALUES  
(1, 2018, 100),
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
(7, 2020, 30),  
(13, 2019, 40),  
(1, 2019, 113),  
(2, 2008, 121),  
(3, 2009, 12),  
(11, 2020, 99),  
(7, 2019, 0);
```

INSERT INTO Queries (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 )  
from Queries as Q  
left outer join  
Year_tbl as Y  
on  
Q.ID=Y.ID and Q.YEAR=Y.YEAR
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## OUTPUT:

ID	YEAR	NVL(Y.NPV,0)
7	2020	30
13	2019	40
1	2019	113
2	2008	121
3	2009	12
7	2019	0
7	2018	0