



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment No: 2

Student Name: Mohammad Kaif

Branch: B.E./C.S.E.

Semester: 5th

Subject Name: ADBMS

Subject Code: 23CSP-333

UID: 23BCS12884

Section/Group: KRG_1-B

Question 1 : Medium-Level Problem

Problem Title : 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,  
    EmpName VARCHAR(50) NOT NULL,  
    Department VARCHAR(50) NOT NULL,  
    ManagerID INT NULL  
);
```

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

```
INSERT INTO Employee VALUES (1, 'Alice', 'Finance', NULL);  
INSERT INTO Employee VALUES (2, 'Bob', 'HR', 1);  
INSERT INTO Employee VALUES (3, 'Charlie', 'Finance', 1);  
INSERT INTO Employee VALUES (4, 'David', 'Research', 2);  
INSERT INTO Employee VALUES (5, 'Eva', 'HR', 2);
```

```
SELECT  
    E1.EmpName AS [EMPLOYEE NAME],  
    E2.EmpName AS [MANAGER NAME],  
    E1.Department AS [EMP_DEPARTMENT],  
    E2.Department AS [MANAGER_DEPT]  
FROM Employee AS E1  
LEFT OUTER JOIN Employee AS E2  
    ON E1.ManagerID = E2.EmpID;
```

Output:

STDIN			
Input for the program (Optional)			
Output:			
EMPLOYEE NAME	MANAGER NAME	EMP_DEPARTMENT	MANAGER_DEPT
Alice	NULL	Finance	NULL
Bob	Alice	HR	Finance
Charlie	Alice	Finance	Finance
David	Bob	Research	HR
Eva	Bob	HR	HR

Question 2 : Hard - Level Problem

Problem Title: NPV Lookup with Missing Data Handling (Medium)

You are a Data Engineer at FinSight Corp, where Net Present Value (NPV) data is stored and queried regularly.

You maintain two tables:

1. Year_tbl — Actual recorded NPV values:
 - ID: Unique Financial instrument identifier
 - YEAR: Year of record
 - NPV: Net Present Value in that year
2. Queries_tbl — Stakeholder NPV queries:
 - ID: Financial instrument identifier
 - YEAR: Year of interest

Task:

1. Create the two tables described above: Year_tbl and Queries_tbl.
2. Insert at least 5–6 rows of data into each
3. Write an SQL query to:
 - Return each ID, YEAR, and the corresponding NPV (if it exists)
 - Replace missing NPV values with 0
 - Order the output by ID and YEAR in ascending order

Solution:

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

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

```
INSERT INTO Y_TBL (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);
```

```
INSERT INTO Q_TBL (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
  Q_TBL Q
LEFT JOIN
  Y_TBL Y
ON
  Q.ID = Y.ID AND Q.YEAR = Y.YEAR
ORDER BY
  Q.ID, Q.YEAR
```

Output:

Input for the program (Optional)

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