Experiment 2.1

Student Name: Ishu Ranjan **UID: 23BCS14216**
Branch: B.E-CSE **Section/Group: KRG-3_A**
Semester: 5th **Date of Performance: 24/07/25**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1. Aim: Organizational Hierarchy Explorer

2. Objective:

- 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.

3. 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, 'Priya', 'Finance', 1),
(3, 'Amit', 'IT', 1),
(4, 'Neha', 'Finance', 2),
(5, 'Raj', 'IT', 3),
(6, 'Meena', '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;
```

Output:

EmployeeName	EmployeeDept	ManagerName	ManagerDept
Ravi	HR	NULL	NULL
Priya	Finance	Ravi	HR
Amit	IT	Ravi	HR
Neha	Finance	Priya	Finance
Raj	IT	Amit	IT
Meena	HR	Ravi	HR

Experiment 2.2

Student Name: Ishu Ranjan **UID: 23BCS14216**
Branch: B.E-CSE **Section/Group: KRG-3_A**
Semester: 5th **Date of Performance: 24/07/25**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1. **Aim:** Financial Forecast Matching with Fallback Strategy

2. **Objective:**
   - 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.

3. **Code:**

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

INSERT INTO YEAR_TABLE(ID,YEAR,NPV) VALUES
(1, 2018, 110),
(1, 2019, 20),
(2, 2008, 130),
(3, 2010, 25),
(4, 2020, 80),
(5, 2021, 90),
```

```
(6, 2020, 0),
(7, 2019, 15),
(7, 2020, 40),
(8, 2017, 50);
CREATE TABLE QUERIES_TABLE( ID INT,
YEAR INT
);
INSERT INTO QUERIES_TABLE( ID,YEAR)
VALUES
(1, 2019),
(2, 2008),
(3, 2009),
(4, 2020),
(5, 2021),
(6, 2020),
(7, 2018),
(7, 2019),
(8, 2017),
(9, 2022);

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	20
2	2008	130
3	2009	0
4	2020	80
5	2021	90
6	2020	0
7	2018	0
7	2019	15
8	2017	50
9	2022	0