Experiment 1.2

Student Name: Naresh Kumar   **UID:** 23BCS13655

Branch: B.E-C.S.E   **Section/Group:** 23BCS_KRG-2B

Semester: 5th   **Date of Performance:** 28 July, 2025

Subject Name: ADBMS   **Subject Code:** 23CSP-333

MEDIUM - LEVEL

1. **Problem Title:** Organizational Hierarchy Explorer

2. **Procedure (Step-by-Step):** 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. **SQL Commands:**
   a. Create the database and use it:
   ```
   CREATE DATABASE VaibhavADBMS;
   USE VaibhavADBMS:
   ```
   b. Create tables StaffDetails:
   ```
   CREATE TABLE StaffDetails (
       StaffID INT IDENTITY(1,1),
       StaffName VARCHAR(MAX),
       Dept VARCHAR(MAX),
       HeadID INT
   );
   ```

c. Insert the values in the tables:

```sql
INSERT INTO StaffDetails (StaffName, Dept, HeadID)
VALUES
('Neha', 'Marketing', NULL),
('Aman', 'Sales', 1),
('Riya', 'Tech', 1),
('Karan', 'Sales', 2),
('Ishita', 'Tech', 3),
('Mohit', 'Marketing', 1);
```

d. Selecting the Employee Name, Manager name, and Employee and Manager Department:

```sql
SELECT S1.StaffName AS [Employee Name],
       S2.StaffName AS [Manager Name],
       S1.Dept AS [Employee Dept],
       S2.Dept AS [Manager Dept]
FROM StaffDetails AS S1
LEFT OUTER JOIN StaffDetails AS S2
ON S1.HeadID = S2.StaffID;
```

5. **Output:**

⊞ Results | 📄 Messages

	Employee Name	Manager Name	Employee Dept	Manager Dept
1	Neha	NULL	Marketing	NULL
2	Aman	Neha	Sales	Marketing
3	Riya	Neha	Tech	Marketing
4	Karan	Aman	Sales	Sales
5	Ishita	Riya	Tech	Tech
6	Mohit	Neha	Marketing	Marketing

6. **Learning Outcome:**
   a. I learned how to create and manage relational databases using SQL.
   b. I understood how to define primary and foreign key constraints to establish relationships between tables.
   c. I learned how to use LEFT OUTER JOIN to retrieve combined data from related tables, including unmatched rows.
   d. I practiced inserting multiple records efficiently into SQL tables.

1. **Problem Title:** Financial Forecast Matching with Fallback Strategy

2. **Procedure (Step-by-Step):** 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.

3. **SQL Commands:**
    a. Create the database and use it:

```sql
CREATE DATABASE VaibhavADBMS;
USE VaibhavADBMS;
```

    b. Create tables InvestmentData and RequestLog:

```sql
CREATE TABLE InvestmentData (
    InstrumentID INT,
    RecordYear INT,
    NetValue INT
);


CREATE TABLE RequestLog (
    InstrumentID INT,
    QueryYear INT
);
```
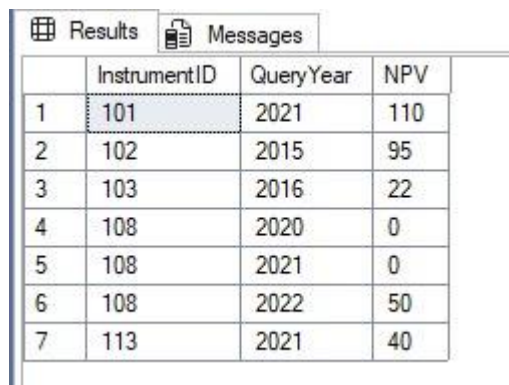
c. Insert the values in the tables:

```sql
INSERT INTO InvestmentData
VALUES
(101, 2020, 75),
(108, 2022, 50),
(113, 2021, 40),
(101, 2021, 110),
(102, 2015, 95),
(103, 2016, 22),
(111, 2022, 17),
(108, 2021, 0);

INSERT INTO RequestLog
VALUES
(101, 2021),
(102, 2015),
(103, 2016),
(108, 2020),
(108, 2021),
(108, 2022),
(113, 2021);
```

d. Selecting the ID, YEAR and NPV:

```sql
SELECT R.InstrumentID, R.QueryYear, ISNULL(I.NetValue, 0) AS NPV
FROM RequestLog AS R
LEFT OUTER JOIN InvestmentData AS I
ON R.InstrumentID = I.InstrumentID AND R.QueryYear = I.RecordYear
ORDER BY R.InstrumentID, R.QueryYear;
```

4. Output:

	InstrumentID	QueryYear	NPV
1	101	2021	110
2	102	2015	95
3	103	2016	22
4	108	2020	0
5	108	2021	0
6	108	2022	50
7	113	2021	40

5. Learning Outcome:

a. I learned how to create and manage relational databases using SQL.

b. I understood how to define primary and foreign key constraints to establish relationships between tables.

c. I practiced inserting multiple records efficiently into SQL tables to populate them with structured data

d. I learned how to use LEFT OUTER JOIN to retrieve combined data from related tables, including unmatched rows from the left table.

e. I learnt how to use ISNULL keyword in SQL queries on tables.