Experiment 2

Student Name: Devansh Sharma | UID: 23BAI70166

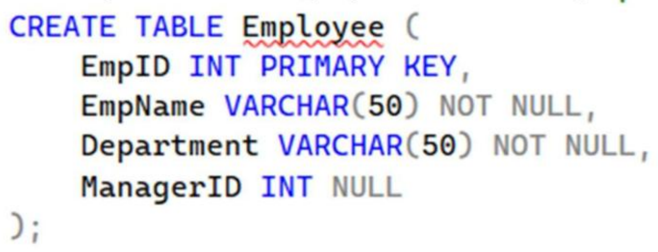
Branch: BE-AIT-CSE | Section/Group: 23AIT_KRG-1 (G1)

Semester: 5th | Date of Performance: 30-07-2025

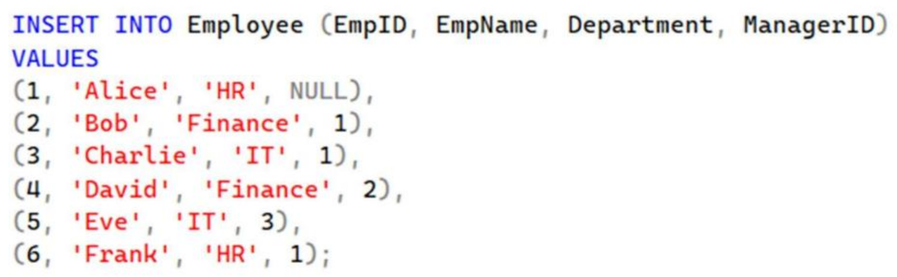
Subject Name: ADBMS | Subject Code: 23CSP-333

MEDIUM - LEVEL

1. Problem Title: Organizational Hierarchy Explorer
2. Problem Tasks and Description:
   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:
   a) The employee's name and department
   b) Their manager's name and department (if applicable)
   c) This will help the HR department visualize the internal reporting hierarchy.
3. SQL Commands:
   a. Creating the table Employee and inserting values into it:

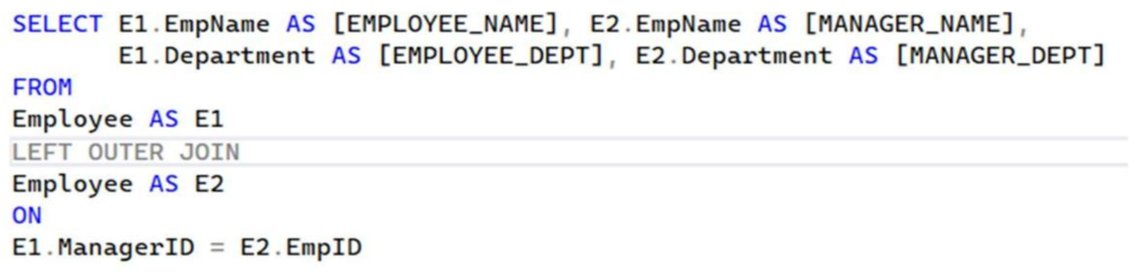


```
CREATE TABLE Employee (
    EmpID INT PRIMARY KEY,
    EmpName VARCHAR(50) NOT NULL,
    Department VARCHAR(50) NOT NULL,
    ManagerID INT NULL
);
```



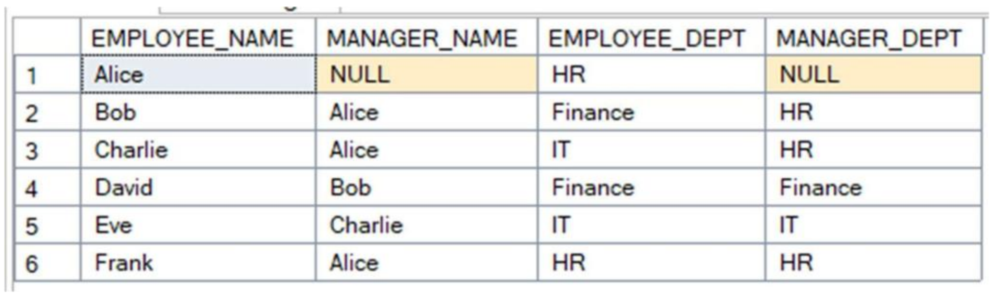
```sql
INSERT INTO Employee (EmpID, EmpName, Department, ManagerID)
VALUES
(1, 'Alice', 'HR', NULL),
(2, 'Bob', 'Finance', 1),
(3, 'Charlie', 'IT', 1),
(4, 'David', 'Finance', 2),
(5, 'Eve', 'IT', 3),
(6, 'Frank', 'HR', 1);
```

b. Using self join to get desired output having the Managers name along with the employee name:



```sql
SELECT E1.EmpName AS [EMPLOYEE_NAME], E2.EmpName AS [MANAGER_NAME],
       E1.Department AS [EMPLOYEE_DEPT], E2.Department AS [MANAGER_DEPT]
FROM
Employee AS E1
LEFT OUTER JOIN
Employee AS E2
ON
E1.ManagerID = E2.EmpID
```

3. Output:



	EMPLOYEE_NAME	MANAGER_NAME	EMPLOYEE_DEPT	MANAGER_DEPT
1	Alice	NULL	HR	NULL
2	Bob	Alice	Finance	HR
3	Charlie	Alice	IT	HR
4	David	Bob	Finance	Finance
5	Eve	Charlie	IT	IT
6	Frank	Alice	HR	HR

Output of Self Join on Employee table

4. Learning Outcome:
    a. I learnt how to create and manage relational databases using SQL.
    b. I learnt how to define primary and foreign key constraints to link tables.
    c. I learnt how to insert multiple records into SQL tables efficiently.
    d. I learnt how to use SELF JOIN to retrieve combined data from the table.

MEDIUM - LEVEL

1. Problem Title: Financial Forecast Matching with Fallback Strategy
2. Problem Task and Description:

   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:

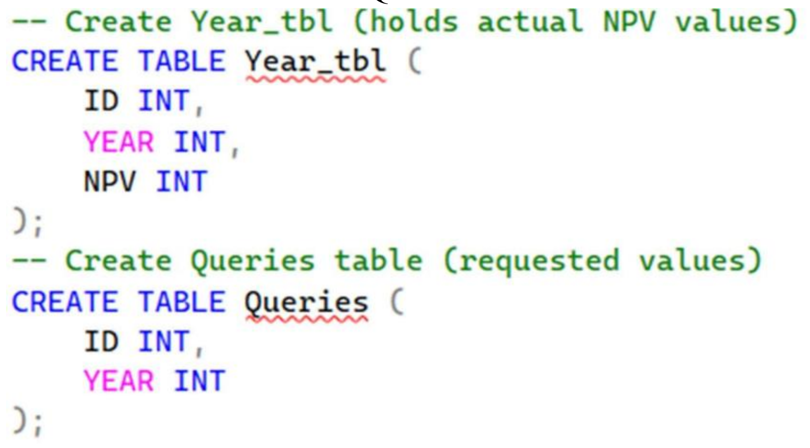
   a) Year_tbl: Actual recorded NPV's of various financial instruments over different years:
      a) ID: Unique Financial instrument identifier.
      b) YEAR: Year of record
      c) NPV: Net Present Value in that year

   b) Queries_tbl: A list of instrument-year pairs for which stakeholders are requesting NPV values:
      a) ID: Financial instrument identifier
      b) YEAR: Year of interest.

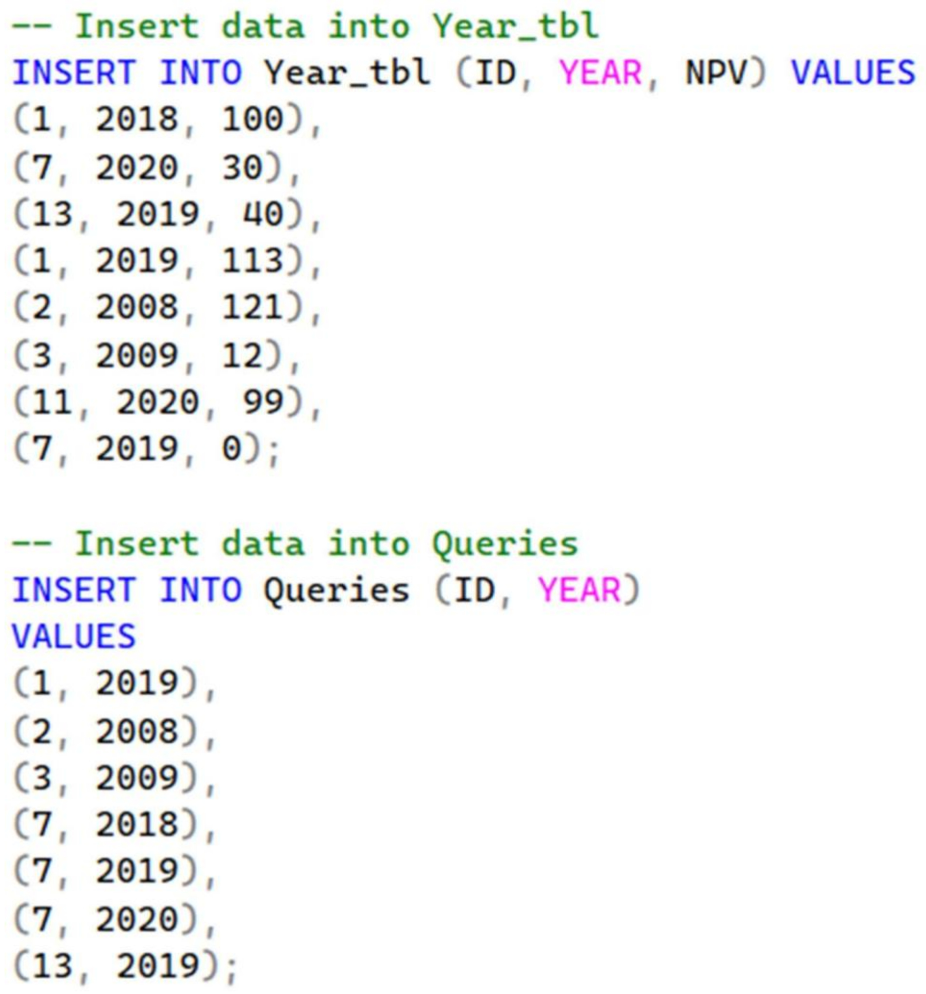
   Find the NPV of each query from the Queries table. Return the output order by ID and Year in the sorted form.

   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 tables Year and Queries and insert the values into them.

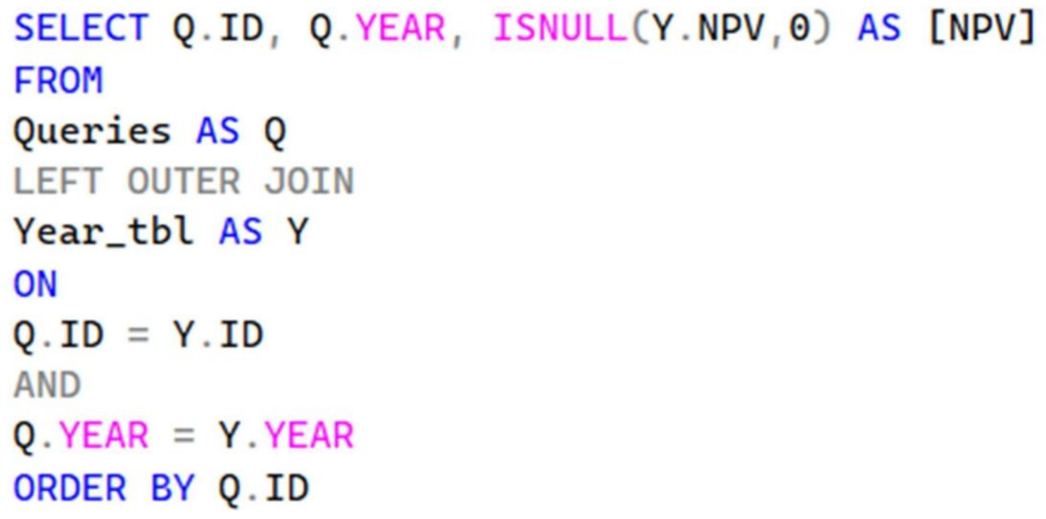


```
-- Create Year_tbl (holds actual NPV values)
CREATE TABLE Year_tbl (
    ID INT,
    YEAR INT,
    NPV INT
);
-- Create Queries table (requested values)
CREATE TABLE Queries (
    ID INT,
    YEAR INT
);
```



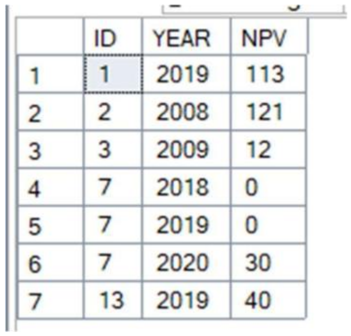
```sql
-- Insert data into Year_tbl
INSERT INTO Year_tbl (ID, YEAR, NPV) VALUES
(1, 2018, 100),
(7, 2020, 30),
(13, 2019, 40),
(1, 2019, 113),
(2, 2008, 121),
(3, 2009, 12),
(11, 2020, 99),
(7, 2019, 0);

-- Insert data into Queries
INSERT INTO Queries (ID, YEAR)
VALUES
(1, 2019),
(2, 2008),
(3, 2009),
(7, 2018),
(7, 2019),
(7, 2020),
(13, 2019);
```

b. Use Left outer join to get desired output and use ORDER BY clause to order the result by the ids.



```sql
SELECT Q.ID, Q.YEAR, ISNULL(Y.NPV,0) AS [NPV]
FROM
Queries AS Q
LEFT OUTER JOIN
Year_tbl AS Y
ON
Q.ID = Y.ID
AND
Q.YEAR = Y.YEAR
ORDER BY Q.ID
```

4. Output:



	ID	YEAR	NPV
1	1	2019	113
2	2	2008	121
3	3	2009	12
4	7	2018	0
5	7	2019	0
6	7	2020	30
7	13	2019	40

Output of Join of Year and Queries table with use of order by clause

5. Learning Outcomes:

    a. Learned about the use of Joins for getting desired output
    b. Learned about the use of Order by Clauses
    c. Learned about the use of aliases for optimization while Querying