Experiment 2

Student Name: Shubh Rai **UID: 23BCS12916**

Branch: B.E.CSE **Section/Group: 23BCS-KRG-3B**

Semester: 5th **Date of Performance:28-07-25**

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

1. Aim:

a) 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.

b) 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.

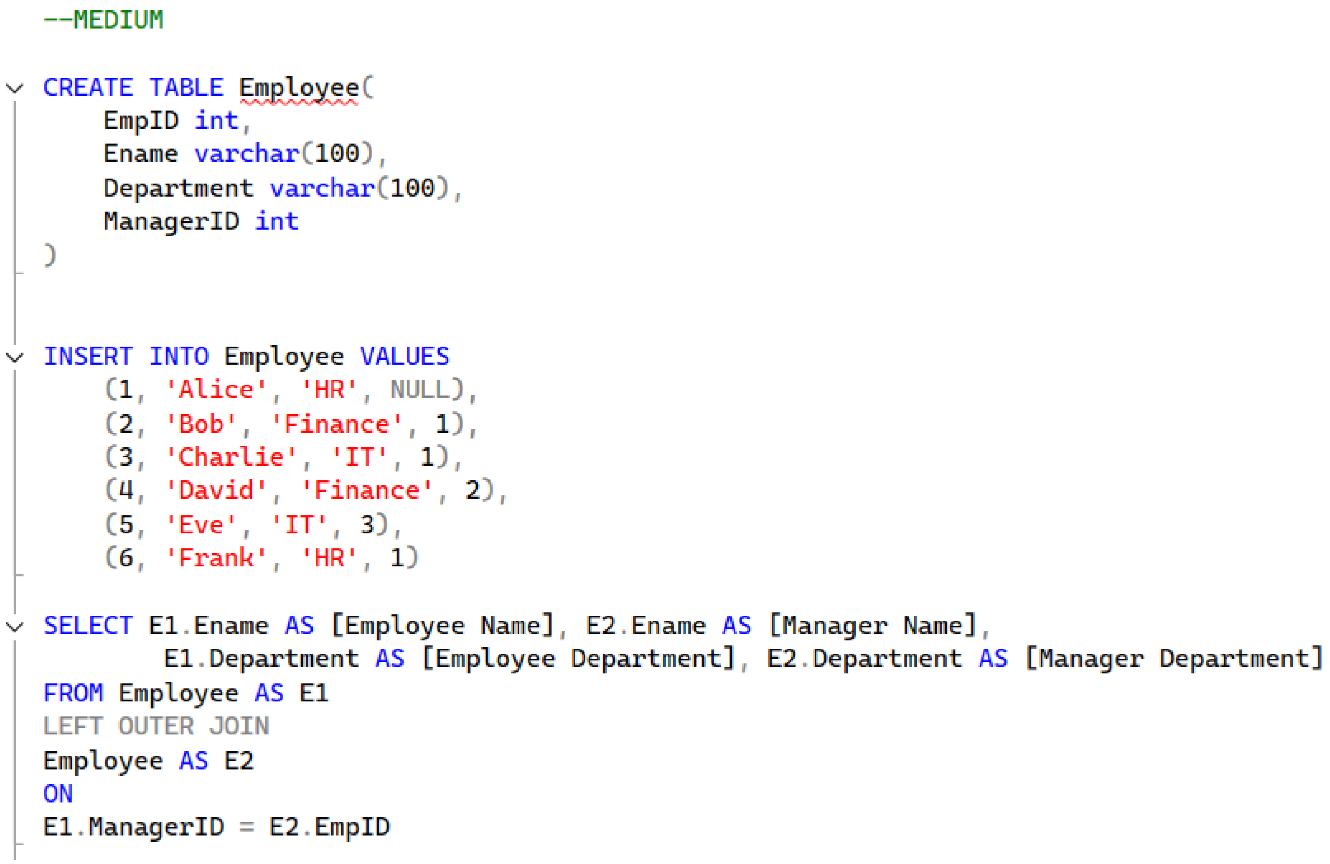
Find the NPV of each query from the Queries table. Return the output order by ID and Year in the sorted form.

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.

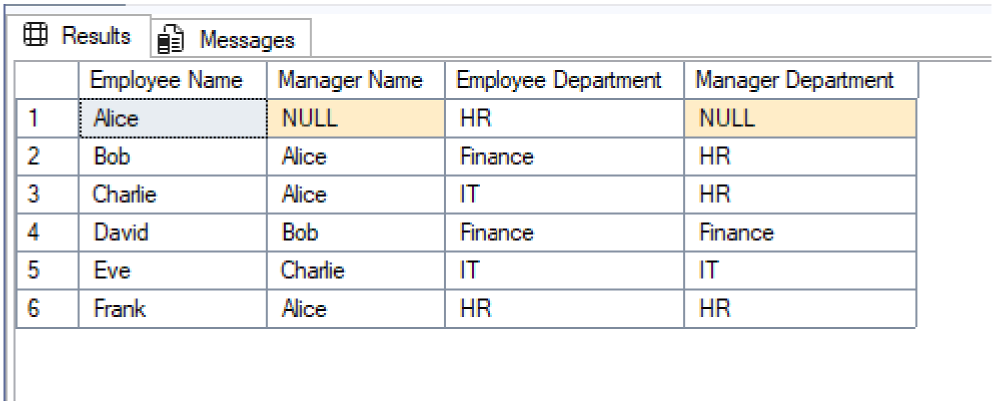
2. Objective:

- To understand how to use JOINS in SQL.
- To understand the basic SQL Queries.
- To generate hierarchical reports from self-referencing tables.

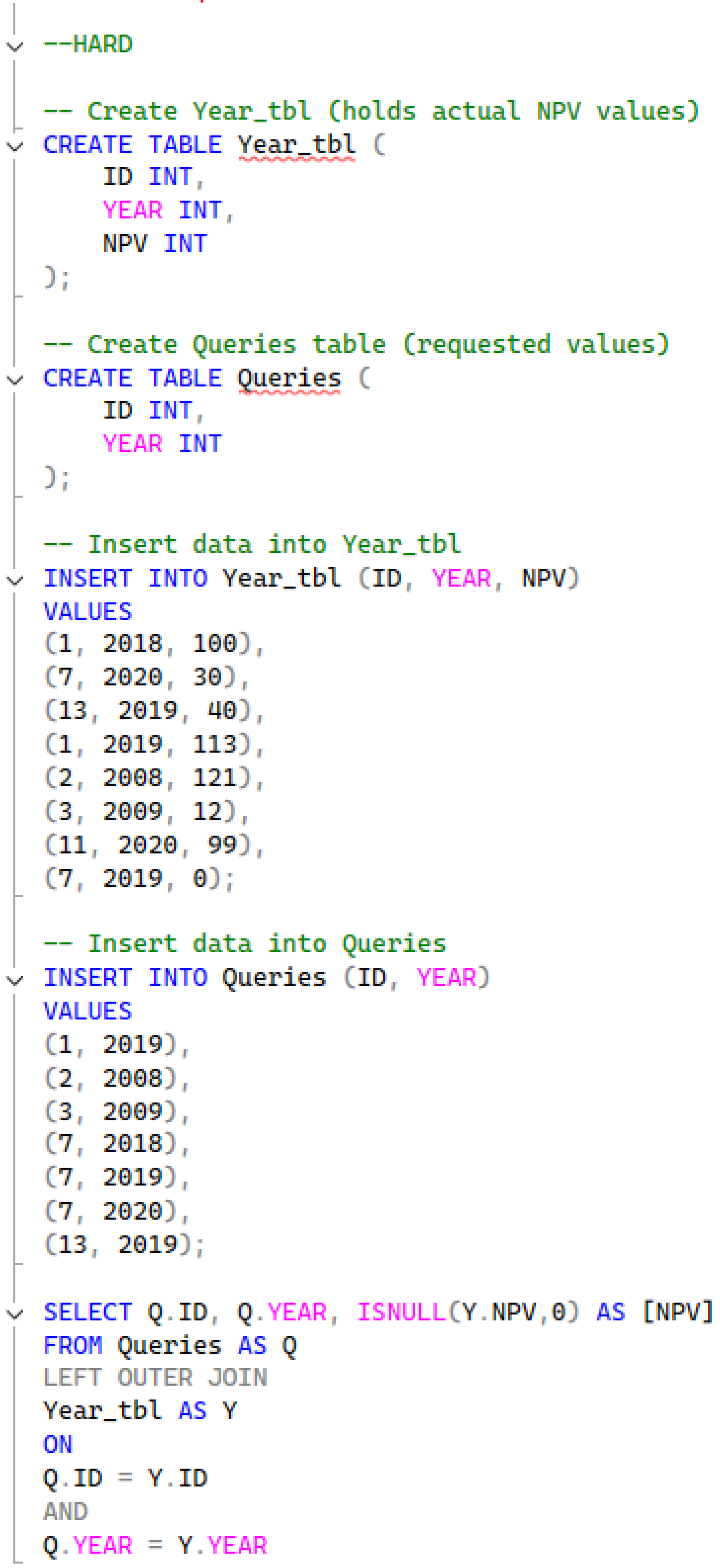
3. DBMS script and output:



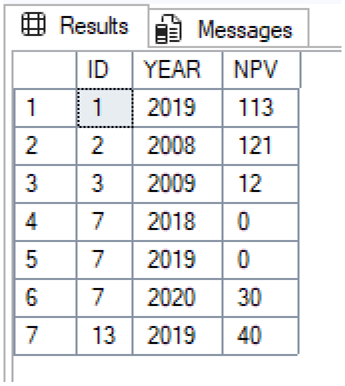
```
--MEDIUM

CREATE TABLE Employee(
    EmpID int,
    Ename varchar(100),
    Department varchar(100),
    ManagerID int
)


INSERT INTO Employee 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)

SELECT E1.Ename AS [Employee Name], E2.Ename AS [Manager Name],
        E1.Department AS [Employee Department], E2.Department AS [Manager Department]
FROM Employee AS E1
LEFT OUTER JOIN
Employee AS E2
ON
E1.ManagerID = E2.EmpID
```



	Employee Name	Manager Name	Employee Department	Manager Department
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



```
--HARD

-- 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
);

-- 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);

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
```



	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

4. Learning outcomes:

- You will be able to write basic SQL queries.
- You will learn to perform JOINS in SQL.
- You will understand how to implement foreign keys.