DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

EXPERIMENT - 2

Student Name: Yash Gupta	**UID: 23BCS11317**
Branch: BE-CSE	**Section/Group: KRG_1-B**
Semester: 5th	**Date of Performance: 29/07/2025**
Subject Name: ADBMS	**Subject Code: 23CSP-333**

1. AIM: Ques 1 :- Organizational Hierarchy Explorer (medium)

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:

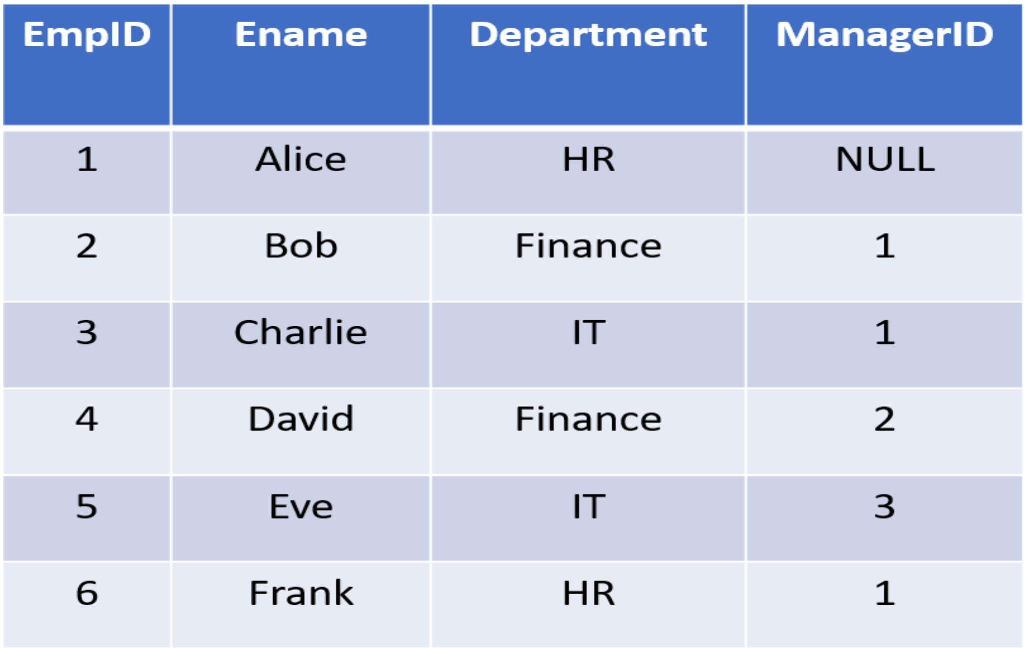
Each employee’s ID, name, department, and manager ID (who is also an employee in the same table).

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.



EmpID	Ename	Department	ManagerID
1	Alice	HR	NULL
2	Bob	Finance	1
3	Charlie	IT	1
4	David	Finance	2
5	Eve	IT	3
6	Frank	HR	1

2. TOOLS USED:- MS SSMS & Microsoft SQL Server

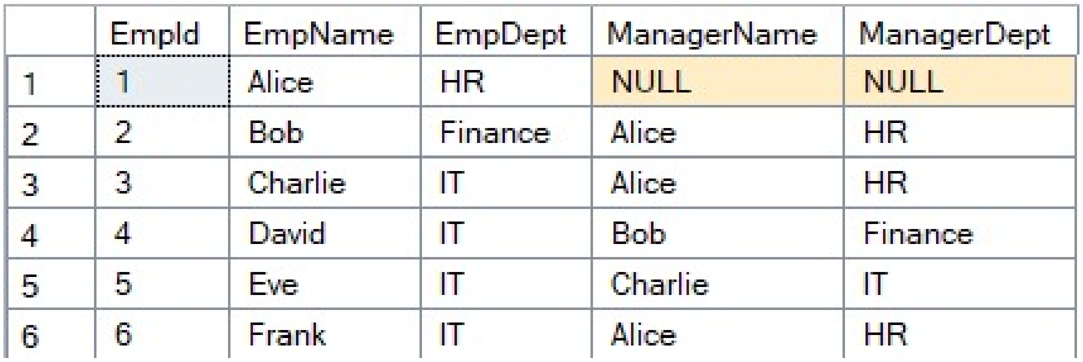
3. SQL CODE:

```
-- create
CREATE TABLE EMPLOYEE (
  empId int primary key,
  name varchar(15),
  dept varchar(10),
  managerId int foreign key references EMPLOYEE(empId)
);

-- insert
INSERT INTO EMPLOYEE VALUES (1, 'Alice', 'HR', NULL);
INSERT INTO EMPLOYEE VALUES (2, 'Bob', 'Finance', 1);
INSERT INTO EMPLOYEE VALUES (3, 'Charlie', 'IT', 1);
INSERT INTO EMPLOYEE VALUES (4, 'David', 'IT', 2);
INSERT INTO EMPLOYEE VALUES (5, 'Eve', 'IT', 3);
INSERT INTO EMPLOYEE VALUES (6, 'Frank', 'IT', 1);

-- fetch
SELECT E.empId as EmpId, E.name as EmpName, E.dept as EmpDept, M.name as
ManagerName, M.dept as ManagerDept
FROM
EMPLOYEE as E
left join
EMPLOYEE as M
on E.managerId = M.empId;
```

4. OUTPUT:



	EmpId	EmpName	EmpDept	ManagerName	ManagerDept
1	1	Alice	HR	NULL	NULL
2	2	Bob	Finance	Alice	HR
3	3	Charlie	IT	Alice	HR
4	4	David	IT	Bob	Finance
5	5	Eve	IT	Charlie	IT
6	6	Frank	IT	Alice	HR

5. Ques 2: -Financial Forecast Matching with Fallback Strategy (hard)

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:

1. 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

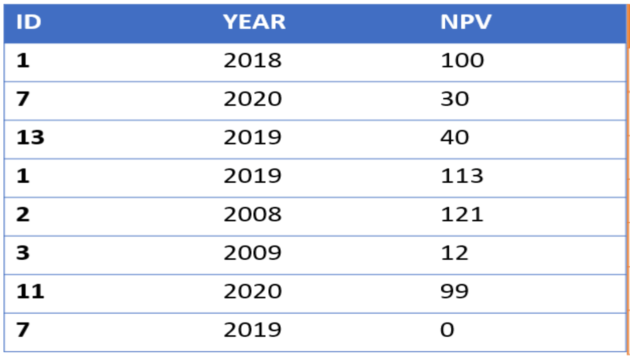
2. Queries_tbl: A list of instrument-year pairs for which stakeholders are requesting NPV values:

   ID: Financial instrument identifier

   YEAR: Year of interest.

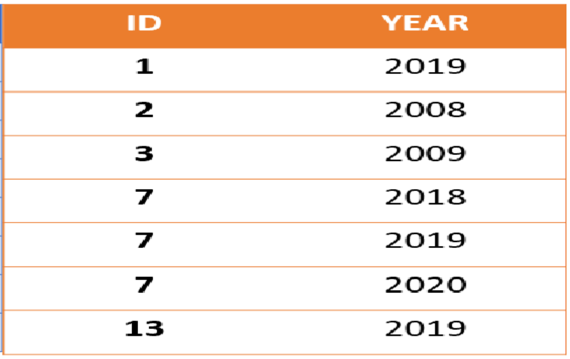
Find the NPV of each query from the Queries table. Return the output order by ID and Year in the sorted form.

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.



ID	YEAR	NPV
1	2018	100
7	2020	30
13	2019	40
1	2019	113
2	2008	121
3	2009	12
11	2020	99
7	2019	0

Year Table



ID	YEAR
1	2019
2	2008
3	2009
7	2018
7	2019
7	2020
13	2019

Queries Table

6. SQL CODE:-

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

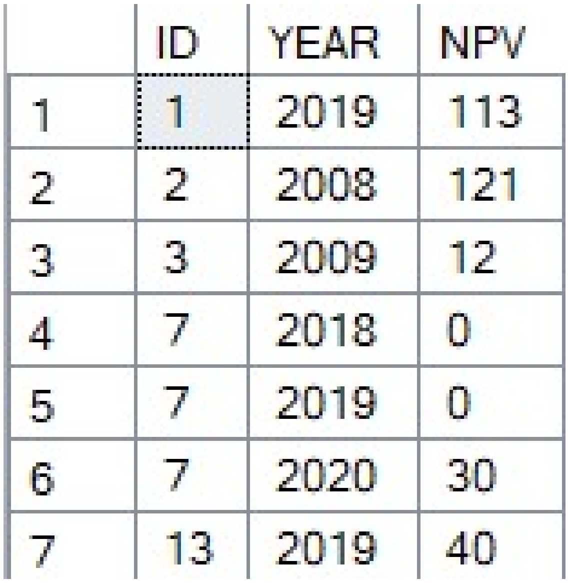
```
-- Create Queries table (requested values)
CREATE TABLE Queries (
    ID INT,
    YEAR INT
);

-- 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.*, ISNULL(Y.NPV, 0) AS NPV
FROM
Year_tbl AS Y
right outer join
Queries AS Q
ON Y.ID=Q.ID AND Y.YEAR = Q.YEAR
ORDER BY Q.ID;
```

7. 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