



University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT : 2

NAME : Raj Kumar Singh

UID: 23BCS11393

BRANCH : BE-CSE

SECTION/GROUP : KRG_3B

SEMESTER : 5TH

SUBJECT CODE : 23CSP-339

SUBJECT NAME : ADBMS

1. Aim Of The Practical :

[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.

[HARD]

To write SQL queries that retrieve requested Net Present Value (NPV) data by performing conditional joins on multiple columns (ID and YEAR) across two related tables, and to handle missing data using appropriate SQL functions.

2. Tools Used : SQL Server Management Studio

3. Code :

----- MEDIUM -----

```
CREATE TABLE TB_EMPLOYEE(  
EMP_ID INT PRIMARY KEY,  
EMP_NAME VARCHAR(25),  
DEPARTMENT VARCHAR(25),
```

```
MANAGER_ID INT);
```

```
INSERT INTO TB_EMPLOYEE  
(EMP_ID,EMP_NAME,DEPARTMENT,MANAGER_ID) 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 * FROM TB_EMPLOYEE;
```

```
SELECT E1.EMP_NAME AS [EMPLOYEE NAME],  
E2.EMP_NAME AS [MANAGER NAME],  
E1.DEPARTMENT AS [EMPLOYEE_DEPT],  
E2.DEPARTMENT AS [MANAGER_DEPT]  
FROM TB_EMPLOYEE AS E1  
LEFT OUTER JOIN  
TB_EMPLOYEE AS E2  
ON  
E1.MANAGER_ID = E2.EMP_ID;
```

----- HARD -----

```
--HARD LEVEL  
--(B) part  
-- 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);
```

```
SELECT * FROM Year_tbl;
```

```
-- 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 * FROM Queries;
```

```
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;
```

4. Output :

[MEDIUM]

	EMP_ID	EMP_NAME	DEPARTMENT	MANAGER_ID
1	1	alice	hr	NULL
2	2	bob	finance	1
3	3	charlie	it	1
4	4	david	finance	2
5	5	eve	it	3
6	6	frank	hr	1

	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

[HARD]

	ID	YEAR	NPV
1	1	2018	100
2	7	2020	30
3	13	2019	40
4	1	2019	113
5	2	2008	121
6	3	2009	12
7	11	2020	99
8	7	2019	0

	ID	YEAR
1	1	2019
2	2	2008
3	3	2009
4	7	2018
5	7	2019
6	7	2020
7	13	2019

	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

5. Learning Outcomes :

- Understand and implement self-joins to model hierarchical relationships within a single table (e.g., employees reporting to other employees).
- Construct relational queries to fetch meaningful information such as employee-manager relationships, including handling NULL values using LEFT JOIN.
- Design and populate tables using the CREATE TABLE and INSERT INTO statements for real-world hierarchical and time-series data scenarios.
- Perform multi-table joins to retrieve and match data across different datasets, such as actual vs. requested values (e.g., NPV values for specific years).
- Handle missing data using functions like ISNULL() to substitute default values during join operations.
- Apply conditional joins involving multiple keys (e.g., joining on both ID and YEAR) to ensure accurate data mapping.
- Develop problem-solving approaches using SQL to derive insights from HR records and financial datasets in enterprise applications.