

Experiment 2

Student Name: Sahil Kumar

UID: 23BCS13733

Branch: BE-CSE

Section/Group: CSE_krg_1_A

Semester: 5th

Date of Performance: 28 July 2025

Subject Name: ADBMS

Subject Code: 23CSP-333

1. Experiment Name:

To design and query a relational database using SQL that involves self-joins for hierarchical relationships, multi-table joins for data lookup, handling missing data with default values, and sorting results for report generation.

2. Objective:

Medium-Level Problem

Problem Title: Employee–Manager Hierarchy Using Self-Join and SQL Retrieval

Procedure (Step-by-Step):

1. Create a single Employee table containing columns: EmpID, Name, Department, and ManagerID (foreign key referencing EmpID in the same table).
2. Insert at least 10 employee records, ensuring some employees have managers and some do not.
3. Use a **self-join** to map each employee to their respective manager based on ManagerID.
4. Select the employee's name and department along with their manager's name and department (if available).
5. Order the results by employee name for clear reporting.

Hard-Level Problem

Problem Title: NPV Lookup with Missing Value Handling and Sorted Output

Procedure (Step-by-Step):

1. Create two tables:
 - Year_tbl with columns: ID, YEAR, NPV.
 - Queries_tbl with columns: ID, YEAR.
2. Insert at least 10 records in Year_tbl and at least 6 in Queries_tbl, ensuring some ID-YEAR pairs in Queries_tbl are **not** present in Year_tbl.
3. Perform a **LEFT JOIN** from Queries_tbl to Year_tbl on both ID and YEAR.
4. Replace NULL NPV values with 0 using ISNULL() or equivalent function.
5. Return the results sorted in ascending order by ID and YEAR.

3. Code:

--MEDIUM---

```
CREATE TABLE EMP (EMPID INT PRIMARY KEY, ENAME VARCHAR(MAX),  
DEPARTMENT VARCHAR(MAX), MANAGERID INT)
```

```
ALTER TABLE EMP
```

```
ADD CONSTRAINT FK_EMP FOREIGN KEY (MANAGERID) REFERENCES  
EMP(EMPID)
```

```
INSERT INTO EMP (EMPID, ENAME, DEPARTMENT, MANAGERID) VALUES
```

```
(101, 'Alice Sharma', 'HR', NULL),  
(102, 'Ravi Kumar', 'Finance', 101),  
(103, 'Priya Singh', 'Engineering', 101),  
(104, 'Vikram Joshi', 'Marketing', 102),  
(105, 'Ankit Verma', 'HR', 102),  
(106, 'Harsh', 'HR', 106)
```

```
SELECT * FROM EMP
```

```
SELECT E1. ENAME AS [EMPLOYEE NAME], E1.DEPARTMENT AS  
[DEPARTMENT] , E2.ENAME AS [MANAGER NAME], E2.DEPARTMENT AS  
[MANAGER DEPT]
```

```
FROM EMP AS E1
```

```
LEFT OUTER JOIN
```

```
EMP AS E2
```

```
ON
```

```
E1.MANAGERID = E2.EMPID
```

---HARD----

```
CREATE TABLE Year_tbl
( ID INT,
  YEAR INT,
  NPV INT
);
```

```
CREATE TABLE Queries
( ID INT,
  YEAR INT
);
```

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

```
SELECT Q.ID AS [ID], Q.YEAR AS [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:

Results		Messages		
	EMPID	ENAME	DEPARTMENT	MANAGERID
1	101	Alice Sharma	HR	NULL
2	102	Ravi Kumar	Finance	101
3	103	Priya Singh	Engineering	101
4	104	Vikram Joshi	Marketing	102
5	105	Ankit Verma	HR	102
6	106	Harsh	HR	106

Results		Messages		
	EMPLOYEE NAME	DEPARTMENT	MANAGER NAME	MANAGER DEPT
1	Alice Sharma	HR	NULL	NULL
2	Ravi Kumar	Finance	Alice Sharma	HR
3	Priya Singh	Engineering	Alice Sharma	HR
4	Vikram Joshi	Marketing	Ravi Kumar	Finance
5	Ankit Verma	HR	Ravi Kumar	Finance
6	Harsh	HR	Harsh	HR

Results		Messages	
	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

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

Results		Messages	
	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** and **foreign key relationships** for hierarchical data within the same table.
- Apply **JOIN operations** (including LEFT JOIN) to combine data from multiple tables and handle missing values using functions like ISNULL().
- Improve skills in **data retrieval and presentation**, including selecting specific fields, aliasing tables, and ordering results.
- Gain practical experience in **designing and querying relational databases** to generate meaningful business reports from raw data.