



## Experiment 2

**Student Name:** Harsh Kumar

**Branch:** CSE

**Semester:** 5<sup>th</sup>

**Subject Name:** ADBMS

**UID:** 23BCS11889

**Section/Group:** KRG\_2A

**Date of Performance:** 31/07/25

**Subject Code:** 23CSP-333

### 1. Aim:

- To design and implement normalized relational database schemas using SQL for employee-manager and financial data scenarios.
- To establish self-referencing and standard foreign key relationships between tables.
- To retrieve specific data using JOIN operations and handle missing values using functions like ISNULL().

#### ❖ Part A– Medium Level:

- To create a table Employee\_tbl to capture employee details along with their reporting manager.
- To implement a self-join on Employee\_tbl to display employee and their manager details.
- To use a LEFT OUTER JOIN to ensure employees without a manager (i.e., top-level managers) are also included in the result.
- To retrieve and display employee name, manager name, and their respective departments.

#### ❖ Part B – Hard Level:

- To create two tables Year\_tbl and Queries for tracking financial Net Present Value (NPV) by year and ID.
- To insert multiple year-wise NPV values for various IDs.
- To query specific combinations of ID and year using a LEFT JOIN.
- To handle missing NPV values using the ISNULL() function, ensuring zero is shown when no record exists.

### 2. Objective:

- To understand and apply self-joins in SQL to relate rows within the same table, such as employee-manager relationships.
- To design and populate relational tables for temporal financial data, enabling time-series queries.
- To retrieve data using outer joins and display comprehensive information, including unmatched rows.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- To utilize SQL functions like ISNULL() for handling NULL values in results.
- To enhance skills in data modeling, joining strategies, and conditional data retrieval in SQL Server.

## 3. ADBMS script and output:

### MEDIUM-LEVEL PROBLEM

```
CREATE TABLE Employee_tbl (  
    EmpID INT,  
    EmpName VARCHAR(100),  
    Dept VARCHAR(100),  
    Manager_ID INT  
);
```

```
INSERT INTO Employee_tbl 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.EmpName AS [EMPLOYEE NAME],  
    E2.EmpName AS [MANAGER NAME],  
    E1.Dept AS [EMPLOYEE DEPARTMENT],
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
E2.Dept AS [MANAGER DEPARTMENT]
FROM
    Employee_tbl AS E1
LEFT OUTER JOIN
    Employee_tbl AS E2
ON
    E1.Manager_ID = E2.EmpID;
```

## **HARD LEVEL PROBLEM:**

```
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),
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

(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,

Q.YEAR,

ISNULL(Y.NPV, 0) AS NPV

FROM

Queries AS Q

LEFT JOIN

Year\_tbl AS Y

ON

Q.ID = Y.ID AND Q.YEAR = Y.YEAR;



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## OUTPUTS:

	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

Figure 1: Medium Level Problem

	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

Figure 2: Hard level Problem