Experiment 2

Student Name:ROHIT KUMAR **UID: 23BCS12640**
Branch: CSE **Section/Group: KRG 3-A**
Semester: 5th **Date of Performance:24/07/2025**
Subject Name: ADBMS **Subject Code: 23CSP-333**

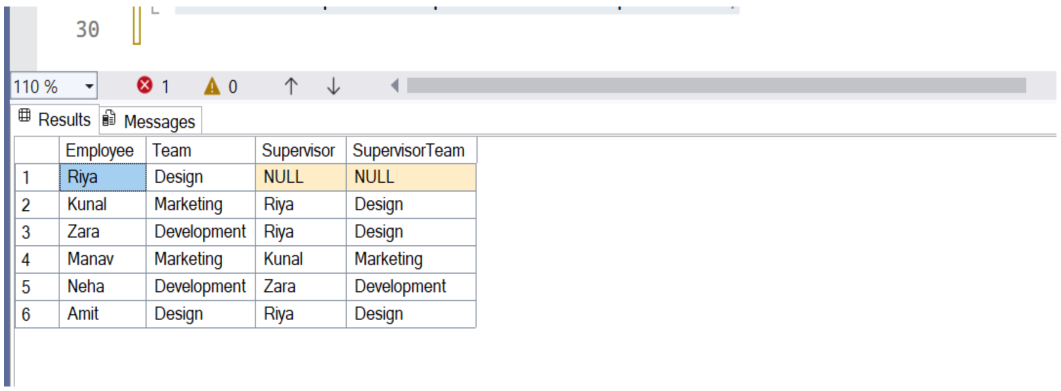
1. **Aim**: To demonstrate the use of self-joins and conditional joins in SQL for managing hierarchical employee relationships and performing conditional lookups using LEFT JOIN and IFNULL across two related tables**.**
   a. Employee-Manager Hierarchy Using Self-Join
   b. Conditional Join Between Financial Tables
2. **Objective:**

- To design and populate relational tables with hierarchical and temporal data.
- To perform a **self-join** on an employee table to retrieve manager-employee relationships.
- To implement a **conditional LEFT JOIN** between two tables to handle non-matching records.
- To apply the **IFNULL** function to handle missing values in joined queries.
- To practice using joins for **querying structured business-related datasets**.

3. DBMS script and output:

Solution-(a)

```
CREATE DATABASE OrgDB;
USE OrgDB;

CREATE TABLE Staff (
    staffId INT PRIMARY KEY,
    staffName VARCHAR(50),
    teamName VARCHAR(50),
    supervisorId INT
);

INSERT INTO Staff (staffId, staffName, teamName, supervisorId) VALUES
(101, 'Riya', 'Design', NULL),
(102, 'Kunal', 'Marketing', 101),
(103, 'Zara', 'Development', 101),
(104, 'Manav', 'Marketing', 102),
(105, 'Neha', 'Development', 103),
(106, 'Amit', 'Design', 101);


SELECT
    s.staffName AS Employee,
    s.teamName AS Team,
    sup.staffName AS Supervisor,
    sup.teamName AS SupervisorTeam
FROM
    Staff s
LEFT JOIN
    Staff sup ON s.supervisorId = sup.staffId;
```

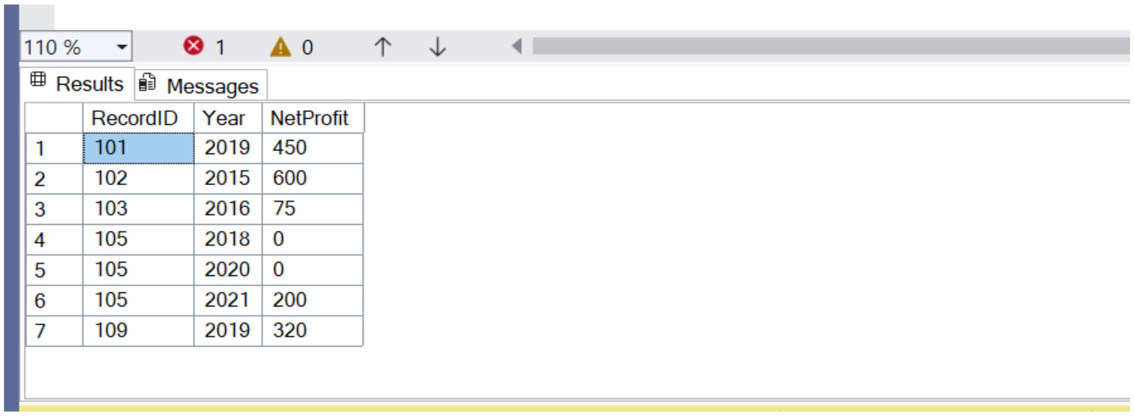


```
110 %   1   0
Results   Messages
```

	Employee	Team	Supervisor	SupervisorTeam
1	Riya	Design	NULL	NULL
2	Kunal	Marketing	Riya	Design
3	Zara	Development	Riya	Design
4	Manav	Marketing	Kunal	Marketing
5	Neha	Development	Zara	Development
6	Amit	Design	Riya	Design

Solution-(b)

```
CREATE DATABASE FinanceDB;
USE FinanceDB;

CREATE TABLE FinancialRecords (
    recordId INT,
    recordYear INT,
    netProfit INT
);

CREATE TABLE LookupRequests (
    requestId INT,
    requestYear INT
);


INSERT INTO FinancialRecords (recordId, recordYear, netProfit) VALUES
(101, 2020, 500),
(105, 2021, 200),
(109, 2019, 320),
(101, 2019, 450),
(102, 2015, 600),
(103, 2016, 75),
(110, 2021, 390),
(105, 2020, 0);


INSERT INTO LookupRequests (requestId, requestYear) VALUES
(101, 2019),
(102, 2015),
(103, 2016),
(105, 2018),
(105, 2020),
(105, 2021),
(109, 2019);


SELECT
    l.requestId AS RecordID,
    l.requestYear AS Year,
    ISNULL(f.netProfit, 0) AS NetProfit
FROM
    LookupRequests l
LEFT JOIN
    FinancialRecords f
ON
    l.requestId = f.recordId AND l.requestYear = f.recordYear;
```



110 % | 1 | 0

Results | Messages

	RecordID	Year	NetProfit
1	101	2019	450
2	102	2015	600
3	103	2016	75
4	105	2018	0
5	105	2020	0
6	105	2021	200
7	109	2019	320

4. Learning Outcomes (What I have Learnt):

- Understand how to model and query **hierarchical relationships** using self-joins.
- Learn to perform **LEFT JOINs** to include unmatched records from one table.
- Apply **composite join conditions** on multiple columns (e.g., ID and YEAR).
- Use **IFNULL** to handle NULL values in result sets for reporting purposes.
- Develop SQL skills for solving **real-world data retrieval scenarios** in organizations.