CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

Advanced Database Management System

Experiment 1.2

23CSP-333

Submitted To:
Faculty Name: Er. Alok Kumar

Submitted By:
Name: Garvi Dabas
UID: 23BCS11346
Section: KRG - 2B
Semester: 5th

Aim:

Problem 1 Title: Employee-Manager Relationship Using Self-Join

Procedure (Step-by-Step):

1. Design a single table TBL_EMPP to store employee information including their manager.
2. Ensure that each employee can optionally have a manager (NULL if top-level).
3. Populate the table with sample employee records and their respective managers.
4. Use a LEFT OUTER JOIN on the same table to retrieve both employee and their manager's details.
5. Display employee name, manager name, and their respective departments.

Sample Output Description:

The query returns each employee's name and department along with their manager's name and department. If an employee has no manager, NULL is shown for the manager's fields.

Problem 2 Title: Query-Based Yearly NPV Lookup Using LEFT JOIN

Procedure (Step-by-Step):

1. Design two tables: one for storing Net Present Value (NPV) entries by year, and the other for querying specific (ID, YEAR) pairs.
2. Insert multiple records into Year_tbl with varied IDs and years along with NPV values.
3. Insert query parameters into the Queries table for which NPV needs to be retrieved.
4. Use a **LEFT OUTER JOIN** to match the records from Queries with Year_tbl based on both ID and year.
5. Use the ISNULL() function to return 0 when no matching NPV is found for a query.

Sample Output Description:

The output shows a list of all queries along with the corresponding NPV. If no matching record exists in Year_tbl, the result displays NPV as 0.

Code:

Answer 1:

```
CREATE TABLE TBL_EMPP(
EMP_ID INT PRIMARY KEY,
ENAME VARCHAR(20),
DEPARTMENT VARCHAR(20),
MANAGER_ID INT)

INSERT INTO TBL_EMPP VALUES(1,'ALICE','HR',NULL)
INSERT INTO TBL_EMPP VALUES(2,'BOB','FINANCE',1)
INSERT INTO TBL_EMPP VALUES(3,'CHARLIE','IT',1)
INSERT INTO TBL_EMPP VALUES(4,'DAVID','FINANCE',2)
INSERT INTO TBL_EMPP VALUES(5,'EVE','IT',3)
INSERT INTO TBL_EMPP VALUES(6,'FRANK','HR',1)
```

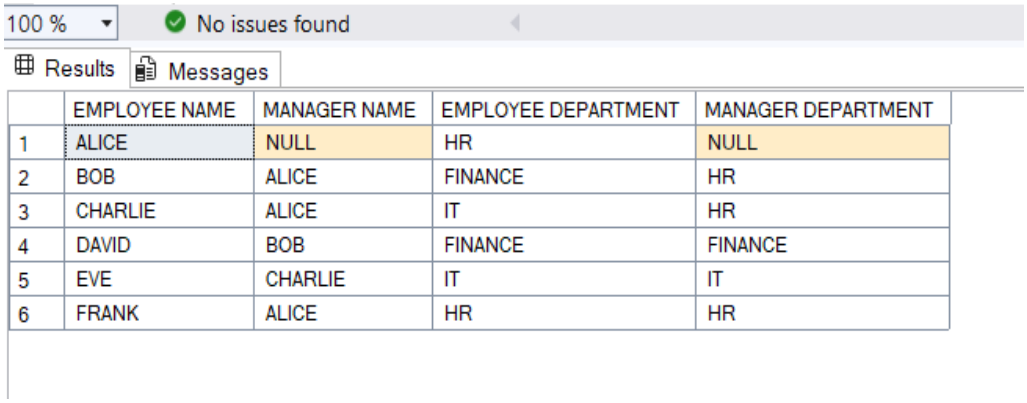
```
SELECT E1.ENAME AS[EMPLOYEE NAME], E2.ENAME AS
[MANAGER NAME],
E1.DEPARTMENT AS[EMPLOYEE DEPARTMENT],
E2.DEPARTMENT AS[MANAGER DEPARTMENT]
FROM TBL_EMPP AS E1
LEFT OUTER JOIN
TBL_EMPP AS E2
ON
E1.MANAGER_ID = E2.EMP_ID
```

Answer 2:

```
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.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
```

Output 1:

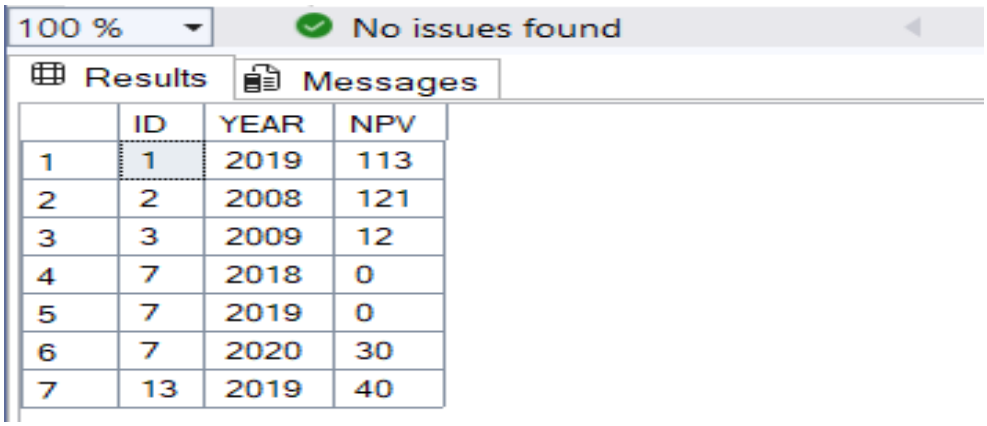


100 % No issues found

Results Messages

	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

Output 2:



100 % No issues found

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

Conclusion :

This experiment demonstrated the use of self-joins and left outer joins to handle hierarchical data and conditional lookups. The first query linked employees with their managers from the same table, while the second ensured all query results were retrieved when matching data was missing, using ISNULL() for default values. These operations are essential for real-world database queries involving relationships and missing data handling.