Experiment 3

Student Name: Deepanshu Mandhyan **UID: 23BCS12327**
Branch: CSE **Section/Group: KRG 3-A**
Semester: 5th **Date of Performance:14/08/2025**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1. Aim:

1. Create an Employee relation with a single attribute EMP_ID. From this relation, find the maximum EMP_ID value after removing duplicates.

2. Define two tables:

a. Department(ID, Name)

b. Employees(ID, Name, Salary, DeptID)

Then, retrieve the employee(s) with the highest salary in each department.

3. Create two tables A(EmpID, EmpName, Salary) and B(EmpID, EmpName, Salary). From both tables, determine the lowest salary of each employee across the two tables..

2. Objective:

- Create employee relation and find max EMP_ID without duplicates.
- Identify highest earners in each department.
- Find lowest salary per employee across two tables.
- Practice SQL queries on creation, grouping, and aggregation.

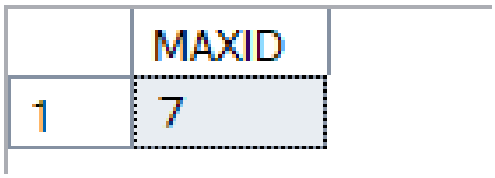
3. DBMS script and output:

Solution-(1)

```
CREATE TABLE EMPLOYEE1 (
   EMP_ID INT
);
```

```
-- Insert given data
INSERT INTO EMPLOYEE1(EMP_ID) VALUES
(2), (4), (4), (6), (6), (7), (8), (8), (8);

SELECT MAX(EMP_ID) AS MAXID FROM EMPLOYEE1 WHERE EMP_ID
NOT IN
(SELECT EMP_ID
FROM
EMPLOYEE1
GROUP BY EMP_ID
HAVING COUNT(EMP_ID)>1 )
```



	MAXID
1	7

Figure(a): JOIN Operation

Solution-(b)

```
-- Create Department Table
CREATE TABLE department (
   id INT PRIMARY KEY,
   dept_name VARCHAR(50)
   );

-- CREATE Employee Table
CREATE TABLE employee (
   id INT,
   name VARCHAR(50),
   salary INT,
   department_id INT,
   FOREIGN KEY (department_id) REFERENCES department(id)
);
```

```
-- Insert into Department Table
INSERT INTO department (id, dept_name) VALUES
(1, 'IT'),
(2, 'SALES');

-- Insert into Employee Table
INSERT INTO employee (id, name, salary, department_id) VALUES
(1, 'JOE', 70000, 1),
(2, 'JIM', 90000, 1),
(3, 'HENRY', 80000, 2),
(4, 'SAM', 60000, 2),
(5, 'MAX', 90000, 1);

-- APROACH 01
SELECT d.dept_name, e.name
FROM department d
INNER JOIN employee e
ON d.id = e.department_id
WHERE e.salary IN
(  SELECT MAX(salary)
   FROM employee
   WHERE department_id = e.department_id
)
ORDER BY d.dept_name
```

	dept_name	name
1	IT	MAX
2	IT	JIM
3	SALES	HENRY

Figure (b): SQL Operation

Solution (c):

```
CREATE TABLE A
( EmpID INT PRIMARY KEY,
  Ename VARCHAR(50),
  Salary INT);
```

```
CREATE TABLE B
( EmpID INT PRIMARY KEY,
  Ename VARCHAR(50),
  Salary INT);

INSERT INTO A (EmpID, Ename, Salary) VALUES
(1, 'AA', 1000),
(2, 'BB', 300);

INSERT INTO B (EmpID, Ename, Salary) VALUES
(2, 'BB', 400),
(3, 'CC', 100);

SELECT EmpID, MIN(Ename), MIN(Salary) AS Salary
FROM
(
SELECT * FROM A
UNION ALL
SELECT * FROM B
) AS Intermediate_result
GROUP BY EmpID
```

	EmpID	(No column name)	Salary
1	1	AA	1000
2	2	BB	300
3	3	CC	100

4. Learning Outcomes:

- Learn to model hierarchies using self-joins
- Use LEFT JOINs to include unmatched rows
- Apply multi-column join conditions (e.g., ID & YEAR)
- Handle NULLs using IFNULL (or ISNULL in SQL Server)
- Build SQL skills for real-world data queries