DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Experiment 3

Student Name: Palash Mathur

UID: 23BAI70673

Branch: BE-AIT-CSE

Section/Group: 23AIT-KRG-G2

Semester: 5th

Date of Performance: 19th Aug, 2025

Subject Name: ADBMS

Subject Code: 23CSP-333

1. **AIM:** In a bustling corporate organization, each department strives to retain the most talented (and well-compensated) employees. You have access to **two key records:** one lists every employee along with their **salary & department**, while the other details the **names of each department**.
   Your task is to identify the top earners in every department. If multiple employees share the same highest salary within a department, all of them should be celebrated equally. The final result should present the department name, employee name, and salary of these top-tier professionals arranged by department.

2. **Tools Used:**
   MySQL on VS CODE.

3. **Experiment:**

   Easy – Level
   Your task is to identify the top earners in every department. If multiple employees share the same highest salary within a department, all of them should be celebrated equally. The final result should present the department name, employee name, and salary of these top-tier professionals arranged by department.

   Medium – Level
   Two legacy HR systems (A and B) have separate records of employee salaries. These records may overlap. Management wants to merge these datasets and identify each unique employee (by EmpID) along with their lowest recorded salary across both systems.

   Objective
   1. Combine two tables A and B.
   2. Return each EmpID with their lowest salary, and the corresponding Ename.

4. Solution:

Easy – Level

```sql
  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 (id, dept_name) VALUES
(1, 'IT'),
(2, 'SALES');

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);


--1. TO JOIN TABLES

SELECT E.*, D.* FROM Employee as E INNER JOIN department as D ON E.department_id
= D.id WHERE (
    SELECT MAX(salary) from Employee WHERE department_id = E.department_id
) ORDER BY D.dept_name;

select d.dept_name, e.name, e.salary
 from employee e INNER JOIN department d on e.department_id = d.id  where salary
in
 (
        select max(e2.salary)
            from employee e2 group by e2.department_id
 )  ORDER BY D.dept_name;
```

Hard-Level

```
CREATE TABLE TABLE_A(EmpID INT(20) PRIMARY KEY, Ename VARCHAR(5), salary
int(20));

CREATE TABLE TABLE_B(EmpID INT(20) PRIMARY KEY, Ename VARCHAR(5), salary
int(20));

INSERT INTO TABLE_A VALUES(1, "AA", 1000), (2, "BB", 300);

INSERT INTO TABLE_B VALUES(2, "BB", 400), (3, "CC", 100);

SELECT EmpID, Ename, salary from (SELECT * FROM TABLE_A UNION ALL SELECT * FROM
TABLE_B) AS RESULT;
```

5. Output:

Easy – Level

dept_name	name	salary
IT	JIM	90000
IT	MAX	90000
SALES	HENRY	80000

Medium – Level

EmpID	Ename	salary
1	AA	1000
2	BB	300
2	BB	400
3	CC	100

6. Learning Outcomes:

- Learn’t about Sub-Queries and it’s types.
- Learn’t about their use cases and execution procedure.
- Lean’t about how to apply Common Table Expression (WITH).
- Understood how to deal with large datasets.