**Experiment-3**

|  |  |
| --- | --- |
| **Student Name:** Manjot Singh | **UID:** 23BCS12549 |
| **Branch:** CSE | **Section/Group:** KRG-2B |
| **Semester:** 5th | **Date of Performance:** 19/08/25 |
| **Subject Name:** ADBMS | **Subject Code:** 23CSP-333 |

1. **Aim:**

**Department Salary Champions Explorer**

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 and 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.

**Merging Employee Histories: Who Earned Least? (Hard)**

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.**

1. **Objective:** 
   * To understand and implement sub-queries in SQL.
   * To identify the top earners in each department using correlated sub-queries.
   * To practice handling scenarios where multiple employees share the same maximum salary.
   * To merge datasets from multiple sources using UNION ALL.
   * To apply GROUP BY with aggregate functions (MAX, MIN) for meaningful reporting.
   * To retrieve the lowest recorded salary for each employee across different systems.
   * To develop practical problem-solving skills for analytical database queries.
2. **DBMS Script :**

USE KRG\_2B;  
  
--EXPERIMENT 03: Department Salary Champions Explorer (MEDIUM LEVEL)  
CREATE TABLE department (  
 id INT PRIMARY KEY,  
 dept\_name VARCHAR(50)  
);  
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);  
  
SELECT (SELECT dept\_name FROM department d where d.id = e.department\_id) AS DEPT\_NAME, name, salary  
FROM Employee e  
WHERE salary IN (SELECT MAX(e2.salary) FROM employee e2 WHERE e2.department\_id = e.department\_id);

--EXPERIMENT 03: Merging Employee Histories: Who Earned Least? (Hard)  
  
CREATE TABLE A( empid integer, Ename VARCHAR(20), Salary INTEGER);  
CREATE TABLE B(empid integer, Ename VARCHAR(20), Salary INTEGER);  
  
INSERT INTO A VALUES  
(1,'AA',1000),  
(2,'BB',300);  
  
INSERT INTO b VALUES  
(2,'BB',400),  
(3,'CC',100);  
  
SELECT EMPID,Max(ENAME) AS ENAME,MIN(SALARY) AS SALARY  
FROM(  
SELECT \* FROM A   
UNION ALL   
SELECT \* FROM B  
) AS INTERMEDIATE\_RESULT  
GROUP BY empid;

1. **Output:**

**Output 1:**

A screenshot of a computer

AI-generated content may be incorrect.

**Output 2:**

A screenshot of a computer

AI-generated content may be incorrect.

1. **Learning Outcomes:**

* Successfully implemented sub-queries to extract top salary earners by department.
* Practiced combining two datasets with UNION ALL.
* Used GROUP BY and aggregate functions (MAX, MIN) to derive meaningful insights.
* Understood how to merge historical records and identify minimum salaries.
* Strengthened SQL querying skills for analytical use cases.