University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT : 2

NAME : SARTHAK UID: 23BCS11978

BRANCH : BE-CSE SECTION/GROUP : KRG_2A

SEMESTER : 5TH SUBJECT CODE : 23CSP-339

SUBJECT NAME : ADBMS

1. Aim Of The Practical :

[MEDIUM]

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 result should present the department name, employee name, and salary of these top-tier professionals arranged by department

[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 EmplD) along with their lowest recorded salary across both systems. Objective: l. Combine two tables A and B. 2. Return each EmplD with their lowest salary, and the corresponding Ename.

2. Tools Used : SQL Server Management Studio

3. Code :

---------- MEDIUM ----------

```
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);

--A1
SELECT D.dept_name, E.name, E.salary
FROM employee AS E
INNER JOIN department AS D
   ON D.id = E.department_id
WHERE E.salary IN (
   SELECT MAX(E2.salary)
   FROM employee AS E2
   WHERE E2.department_id = E.department_id
)
ORDER BY D.dept_name, E.name;

--A2
SELECT D.dept_name, E.name, E.salary
FROM employee AS E
INNER JOIN department AS D
   ON D.id = E.department_id
WHERE E.salary IN (
   SELECT MAX(E2.salary)
   FROM employee AS E2
   Group By E2.department_id
)
ORDER BY D.dept_name, E.name;

---------- HARD ----------
CREATE TABLE A (
   EmpID  INT,
   Ename VARCHAR(50),
```

```
    Salary INT,
);
CREATE TABLE B (
    EmpID  INT,
    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', 1000),
(3, 'CC', 300);

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

4. Output :

[MEDIUM]

	dept_name	name	salary
1	IT	JIM	90000
2	IT	MAX	90000
3	SALES	HENRY	80000

	dept_name	name	salary
1	IT	JIM	90000
2	IT	MAX	90000
3	SALES	HENRY	80000

[HARD]

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

5. Learning Outcomes :
    - Understand and implement self-joins to model hierarchical relationships within a single table (e.g., employees reporting to other employees).
    - Construct relational queries to fetch meaningful information such as employeemanager relationships, including handling NULL values using LEFT JOIN.
    - Design and populate tables using the CREATE TABLE and INSERT INTO statements for real-world hierarchical and time-series data scenarios.
    - Perform multi-table joins to retrieve and match data across different datasets, such as actual vs. requested values (e.g., NPV values for specific years).
    - Handle missing data using functions like ISNULL() to substitute default values during join operations.
    - Apply conditional joins involving multiple keys (e.g., joining on both ID and YEAR) to ensure accurate data mapping.
    - Develop problem-solving approaches using SQL to derive insights from HR records and financial datasets in enterprise applications.