EXPERIMENT - 3

Student Name: Yash Kotra **UID: 23BCS10599**
Branch: BE-CSE **Section/Group: KRG_2A**
Semester: 5th
Subject Name: ADBMS **Subject Code: 23CSP-333**

1. **AIM: Ques 1** :- Create a table dept (id, Dept_Name) and a table MyEmployees (EmpId, EmpName, Gender, Salary, City, Dept_id with foreign key referencing dept). Insert suitable records into both tables. Write an SQL query to find the second highest salary from the MyEmployees table without using TOP or LIMIT **(Easy Level)**

2. **TOOLS USED:-** MS SSMS & Microsoft SQL Server

3. **SQL CODE:**

```
CREATE TABLE MyEmployees (
  EmpId INT PRIMARY KEY IDENTITY(1,1),
  EmpName VARCHAR(50),
  Gender VARCHAR(10),
  Salary INT,
  City VARCHAR(50),
  Dept_id INT
);
INSERT INTO MyEmployees (EmpName, Gender, Salary, City, Dept_id)
VALUES
('Amit', 'Male', 50000, 'Delhi', 2),
('Priya', 'Female', 60000, 'Mumbai', 1),
('Rajesh', 'Male', 45000, 'Agra', 3),
('Sneha', 'Female', 55000, 'Delhi', 4),
('Anil', 'Male', 52000, 'Agra', 2),
```

```
('Sunita', 'Female', 48000, 'Mumbai', 1),
('Vijay', 'Male', 47000, 'Agra', 3),
('Ritu', 'Female', 62000, 'Mumbai', 2),
('Alok', 'Male', 51000, 'Delhi', 1),
('Neha', 'Female', 53000, 'Agra', 4),
('Simran', 'Female', 33000, 'Agra', 3);

create table dept(
id int unique not null,
Dept_Name varchar(20) not null
)
insert into dept values(1, 'Accounts');
insert into dept values(2, 'HR');
insert into dept values(3, 'Admin');
insert into dept values(4, 'Counselling');

SELECT *FROM MyEmployees
SELECT MAX(SALARY) AS [2ND_HIGHEST] FROM MyEmployees WHERE SALARY !=
(SELECT MAX(SALARY) FROM MyEmployees) --62000
```

4. OUTPUT:

	2ND_HIGHEST
1	60000

5. Ques 2: -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. **(Medium Level)**

Employee Table

ID	NAME	SALARY	DEPT_ID
1	JOE	70000	1
2	JIM	90000	1
3	HENRY	80000	2
4	SAM	60000	2
4	MAX	90000	1

department table

ID	DEPT_NAME
1	IT
2	SALES

6. SQL CODE:-

```
    CREATE TABLE department (
  id INT PRIMARY KEY,
  dept_name VARCHAR(50)
);

-- Create Employee Table
CREATE TABLE employees (
  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 employees (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 d.dept_name, e.name, e.salary
FROM employees e
JOIN department d ON e.department_id = d.id
WHERE e.salary = (
   SELECT MAX(salary)
   FROM employees
   WHERE department_id = e.department_id
)
 ORDER BY d.dept_name;
```

7. OUTPUT

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

8. Ques 3:-

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. (**Hard Level**)

Objective

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

Table A

EmpID	Ename	Salary
1	AA	1000
2	BB	300

Table B

EmpID	Ename	Salary
2	BB	400
3	CC	100

9. SQL Code:-

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

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

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

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

SELECT EmpID,Ename,min(Salary)
FROM
(SELECT * FROM Table_A
UNION ALL
SELECT * FROM Table_B)
AS INTERMEDIATE_RESULT
GROUP BY EmpID,Ename
```

10. OUTPUT:-

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