EXPERIMENT - 3

Student Name: Abhay Mall **UID: 23BCS13972**
Branch: BE-CSE **Section/Group: KRG_2A**
Semester: 5th
Subject Name: ADBMS **Subject Code: 23CSP-333**

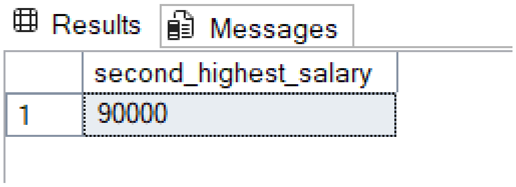
1. **AIM: Ques 1** :- Create a table dept (id, Dept_Name) and a table MyEmployees (EmpId, EmpName, Gender, Salary, City, Dept_id with foreign key referencing dept). Insert suitable records into both tables. Write an SQL query to find the second highest salary from the MyEmployees table without using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

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

3. SQL CODE:

```
create table dept (
  id int primary key,
  dept_name varchar(100)
);

create table myemployees (
  empid int primary key,
  empname varchar(100),
  gender varchar(10),
  salary int,
  city varchar(100),
  dept_id int,
  foreign key (dept_id) references dept(id)
);

insert into dept (id, dept_name) values
  (1, 'HR'),
  (2, 'IT'),
  (3, 'Sales');

insert into myemployees (empid, empname, gender, salary, city, dept_id) values
  (1, 'Alice', 'Female', 70000, 'New York', 2),
  (2, 'Bob', 'Male', 80000, 'Los Angeles', 1),
  (3, 'Charlie', 'Male', 75000, 'Chicago', 3),
  (4, 'David', 'Male', 95000, 'Houston', 2),
  (5, 'Eve', 'Female', 90000, 'San Francisco', 1);

select max(salary) as second_highest_salary
from myemployees
where salary < (select max(salary) from myemployees);
```

4. OUTPUT:



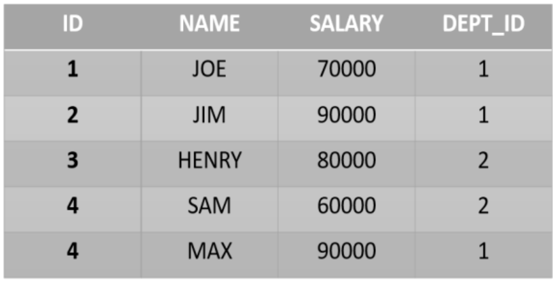
	second_highest_salary
1	90000

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.

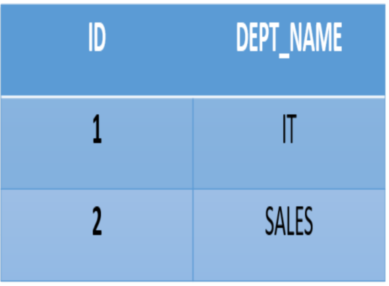
If multiple employees share the same highest salary within a department, all of them should be celebrated equally. The final result should present the department name, employee name, and salary of these top-tier professionals arranged by department. (Medium Level)

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

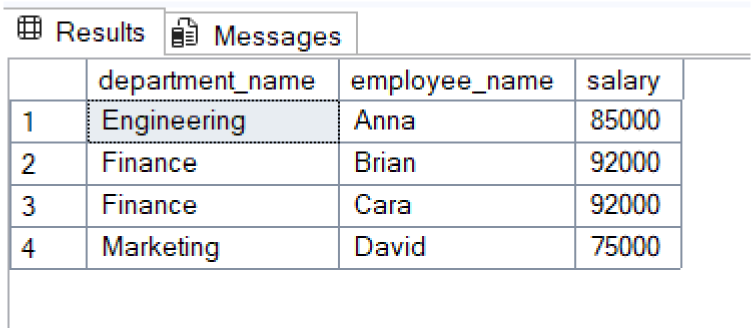


ID	DEPT_NAME
1	IT
2	SALES

6. SQL CODE:-

```
create table departments (
  dept_id int primary key,
  dept_title varchar(100)
);

create table employees (
  employee_id int primary key,
  employee_name varchar(100),
  gender varchar(10),
  salary_amount int,
  city varchar(100),
  department_id int,
  foreign key (department_id) references departments(dept_id)
);

insert into departments (dept_id, dept_title) values
  (1, 'Finance'),
  (2, 'Marketing'),
  (3, 'Engineering');

insert into employees (employee_id, employee_name, gender, salary_amount, city,
department_id) values
  (101, 'Anna', 'Female', 85000, 'Boston', 3),
  (102, 'Brian', 'Male', 92000, 'Seattle', 1),
  (103, 'Cara', 'Female', 92000, 'Austin', 1),
  (104, 'David', 'Male', 75000, 'Denver', 2),
  (105, 'Eva', 'Female', 80000, 'San Diego', 3);

select d.dept_title as department_name,
       e.employee_name as employee_name,
       e.salary_amount as salary
from employees e
inner join departments d on e.department_id = d.dept_id
where e.salary_amount = (
  select max(salary_amount)
  from employees
  where department_id = e.department_id
)
order by d.dept_title;
```

7. OUTPUT



	department_name	employee_name	salary
1	Engineering	Anna	85000
2	Finance	Brian	92000
3	Finance	Cara	92000
4	Marketing	David	75000

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

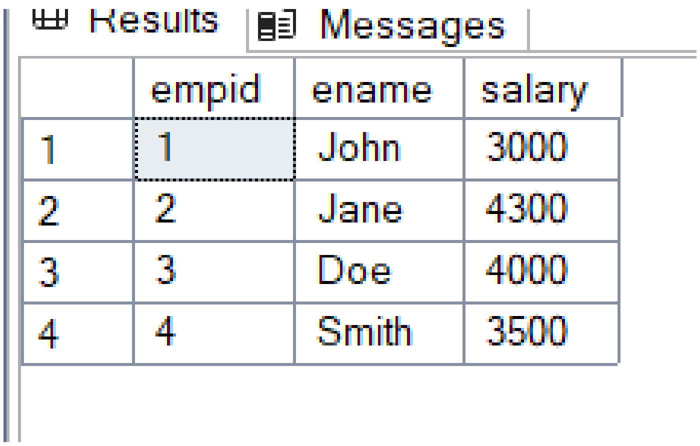
Table B

EmpID	Ename	Salary
2	BB	400
3	CC	100

9. SQL Code:-

```
create table table_a (
  empid int primary key,
  ename varchar(100),
  salary int
);

create table table_b (
  empid int primary key,
  ename varchar(100),
  salary int
);

insert into table_a (empid, ename, salary) values
  (1, 'John', 3000),
  (2, 'Jane', 4500),
  (3, 'Doe', 4000);

insert into table_b (empid, ename, salary) values
  (2, 'Jane', 4300),
  (3, 'Doe', 4200),
  (4, 'Smith', 3500);

select empid, ename, salary
from (
  select empid, ename, salary,
         row_number() over (partition by empid order by salary asc) as rn
  from (
    select * from table_a
    union all
    select * from table_b
  ) combined
) ranked
where rn = 1
order by empid;
```

10. OUTPUT:-



	empid	ename	salary
1	1	John	3000
2	2	Jane	4300
3	3	Doe	4000
4	4	Smith	3500