**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)**

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

1. **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),

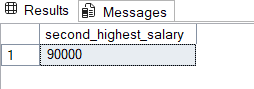
(5, 'Eve', 'Female', 90000, 'San Francisco', 1);

select max(salary) as second\_highest\_salary

from myemployees

where salary < (select max(salary) from myemployees);

1. **OUTPUT:**

****

1. **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

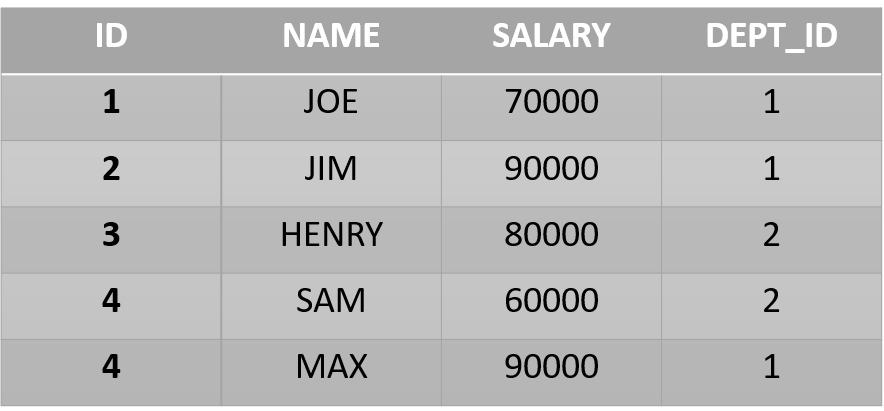
task is to identify the top earners in every department.

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 Department table



# 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)

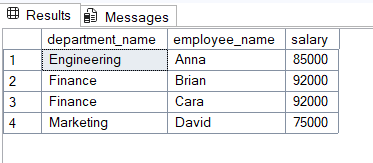
from employees

where department\_id = e.department\_id

)

order by d.dept\_title;

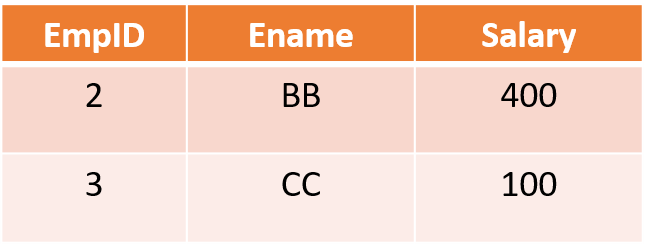
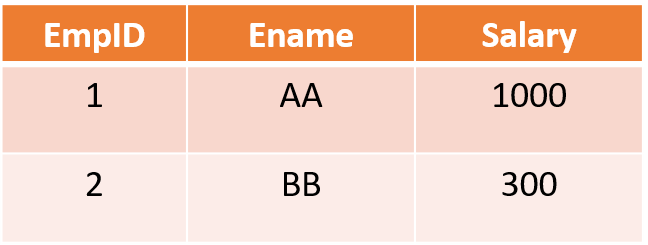
# 7. OUTPUT

****

**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 Table B**



# 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:-

