**Assignment No:-57**

Name:-Suryawanshi Sangramsingh Sambhaji

Batch: - Delta - DCA (Java) 2024 Date:-1/8/2024

CREATE TABLE Employee (

emp\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

department VARCHAR(50),

salary DECIMAL(10, 2),

joining\_date DATE

);

INSERT INTO Employee (first\_name, last\_name, department, salary, joining\_date) VALUES

('Amit', 'Sharma', 'HR', 50000.00, '2020-01-15'),

('Priya', 'Verma', 'Finance', 60000.00, '2019-03-25'),

('Ravi', 'Kumar', 'IT', 75000.00, '2021-06-10'),

('Neha', 'Singh', 'Marketing', 55000.00, '2018-07-20'),

('Rahul', 'Yadav', 'Sales', 45000.00, '2022-02-28'),

('Sakshi', 'Patel', 'IT', 72000.00, '2020-11-12'),

('Anil', 'Rao', 'HR', 48000.00, '2017-05-15'),

('Deepika', 'Reddy', 'Finance', 63000.00, '2019-08-05'),

('Sunil', 'Gupta', 'Marketing', 57000.00, '2018-10-22'),

('Pooja', 'Nair', 'Sales', 46000.00, '2021-01-14');

```

Trigger-Based Questions

1. Create a trigger to automatically update a timestamp column whenever a row in the `Employee` table is updated.

2. Create a trigger to prevent deletion of rows from the `Employee` table if the employee's department is 'IT'.

3. Create a trigger to log salary changes in another table called `SalaryChanges` whenever a row in the `Employee` table is updated.

4. Create a trigger to ensure that the salary cannot be less than 30000 during insertion.

5. Create a trigger that automatically sets the department to 'General' if it is left blank during an insertion.

6. Create a trigger that calculates the total number of employees in each department and stores it in another table called `DepartmentStats`.

7. Create a trigger that sends an email notification whenever a new row is inserted into the `Employee` table.

8. Create a trigger to prevent updates to the `joining\_date` column after a record is inserted.

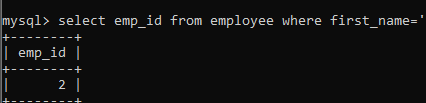
9. Create a trigger that logs the old and new values of the `salary` in a `SalaryAudit` table before an update occurs.

10. Create a trigger to check if an employee with the same first name and last name already exists before inserting a new row.

Subquery-Based Questions

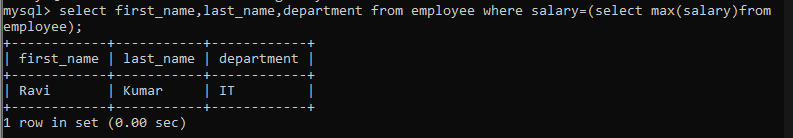
1. Write a subquery to find the `emp\_id` of the employee with the first name 'Priya'.

select emp\_id from employee where first\_name='Priya';



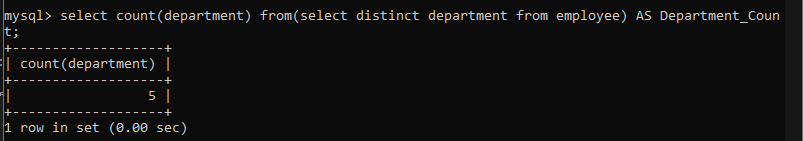
2. Write a subquery to find the full name and department of the employee with the highest salary.

select first\_name,last\_name,department from employee where salary=(select max(salary)from employee);



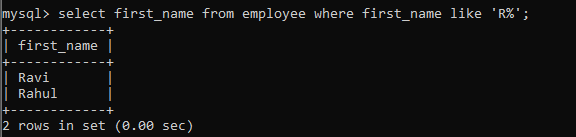
3. Write a subquery to count the number of distinct departments in the `Employee` table.

select count(department) from(select distinct department from employee) AS Department\_Count;



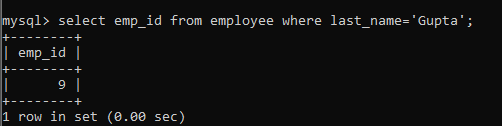
4. Write a subquery to find all rows where the `first\_name` starts with the letter 'R'.

select first\_name from employee where first\_name like 'R%';



5. Write a subquery to retrieve the `emp\_id` of the employee whose last name is 'Gupta'.

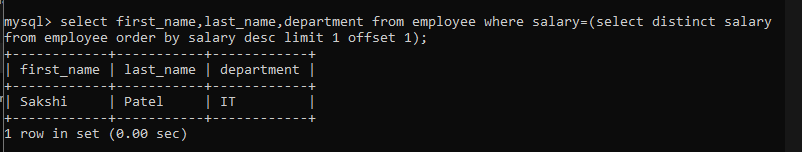
select emp\_id from employee where last\_name='Gupta';



6. Write a subquery to find the `first\_name`, `last\_name`, and `department` of the employee with

the second highest salary.

select first\_name,last\_name,department from employee where salary=(select distinct salary from employee order by salary desc limit 1 offset 1);

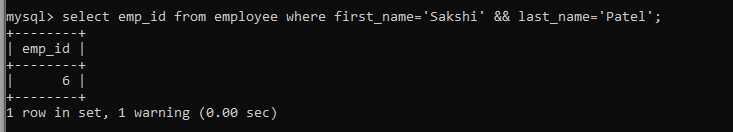


7. Write a subquery to list all `first\_name` values that appear more than once in the `Employee` table.

select first\_name from employee group by first\_name having count(first\_Name)>1;

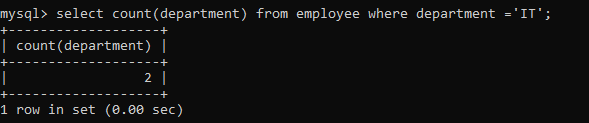


8. Write a subquery to find the `emp\_id` of the row where the `first\_name` is 'Sakshi' and the `last\_name` is 'Patel'.

select emp\_id from employee where first\_name='Sakshi' && last\_name='Patel'; 

9. Write a subquery to find the total number of employees in the 'IT' department.

select count(department) from employee where department ='IT';



10. Write a subquery to list all rows where the `salary` is greater than the average `salary` of all employees in the `Employee` table.

select \* from employee where salary > (select avg(salary) from employee);

