

# ScienceQtech Employee Performance Mapping

## MySQL – Project Write-up

### Objective:

To facilitate a better understanding, managers have provided ratings for each employee which will help the HR department to finalize the employee performance mapping.

### Description:

ScienceQtech is a startup that works in the Data Science field. ScienceQtech has worked on fraud detection, market basket, self-driving cars, supply chain, algorithmic early detection of lung cancer, customer sentiment, and the drug discovery field. With the annual appraisal cycle around the corner, the HR department has asked you (Junior Database Administrator) to generate reports on employee details, their performance, and the project that the employees have undertaken, to analyze the employee database and extract specific data based on different requirements.

### Problem Statement and Motivation:

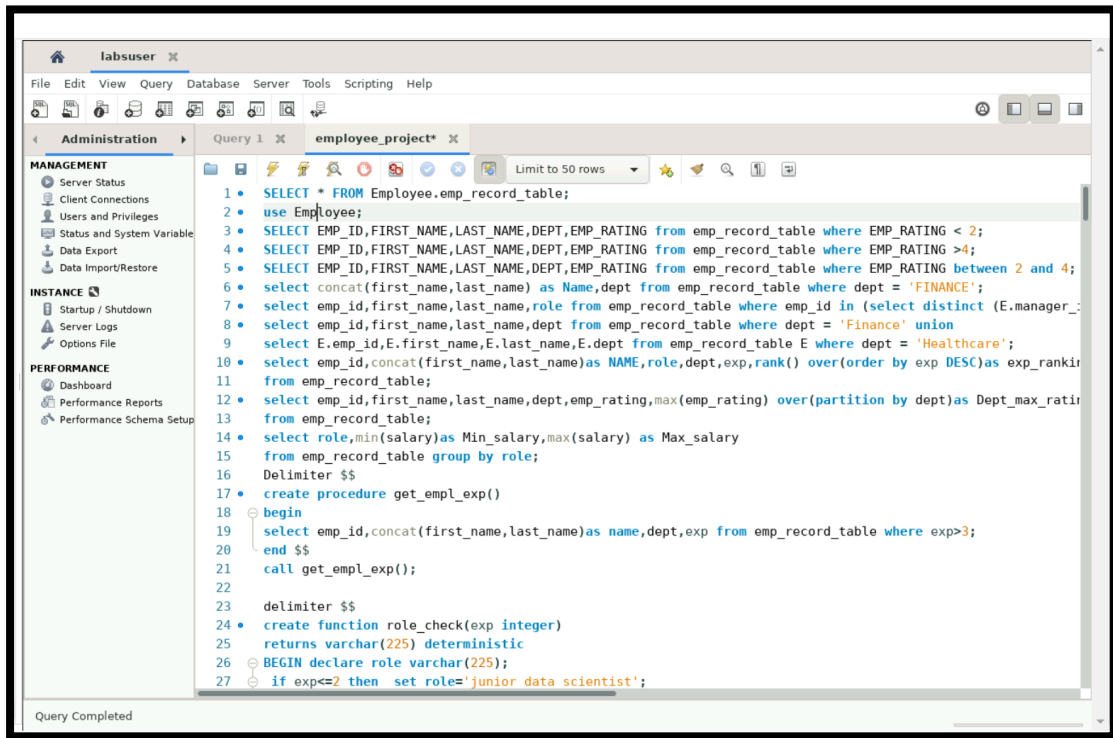
As a DBA (Database Administrator), you should find the maximum salary of the employees and ensure that all jobs are meeting the organization's profile standard. You also need to calculate bonuses to find the extra cost for expenses.

### Tasks:

1. Create a database named employee, then import data\_science\_team.csv, proj\_table.csv and emp\_record\_table.csv into the employee database from the given resources.
2. Create an ER diagram for the given employee database.
3. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department
4. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:
  - less than two
  - greater than four
  - between two and four
5. Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.
6. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).

7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.
8. Write a query to list down employee details such as EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPARTMENT, and EMP\_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department
9. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.
10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.
11. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.
12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.
13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table
14. Write a query using stored functions in the project table to check whether the job profile assigned to each employee in the data science team matches the organization's set standard. The standard is:
  - For an employee with experience less than or equal to 2 years assign 'JUNIOR DATA SCIENTIST',
  - For an employee with the experience of 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST',
  - For an employee with the experience of 5 to 10 years assign 'SENIOR DATA SCIENTIST',
  - For an employee with the experience of 10 to 12 years assign 'LEAD DATA SCIENTIST'
15. Create an index to improve the cost and performance of the query to find the employee whose FIRST\_NAME is 'Eric' in the employee table after checking the execution plan.
16. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).
17. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table

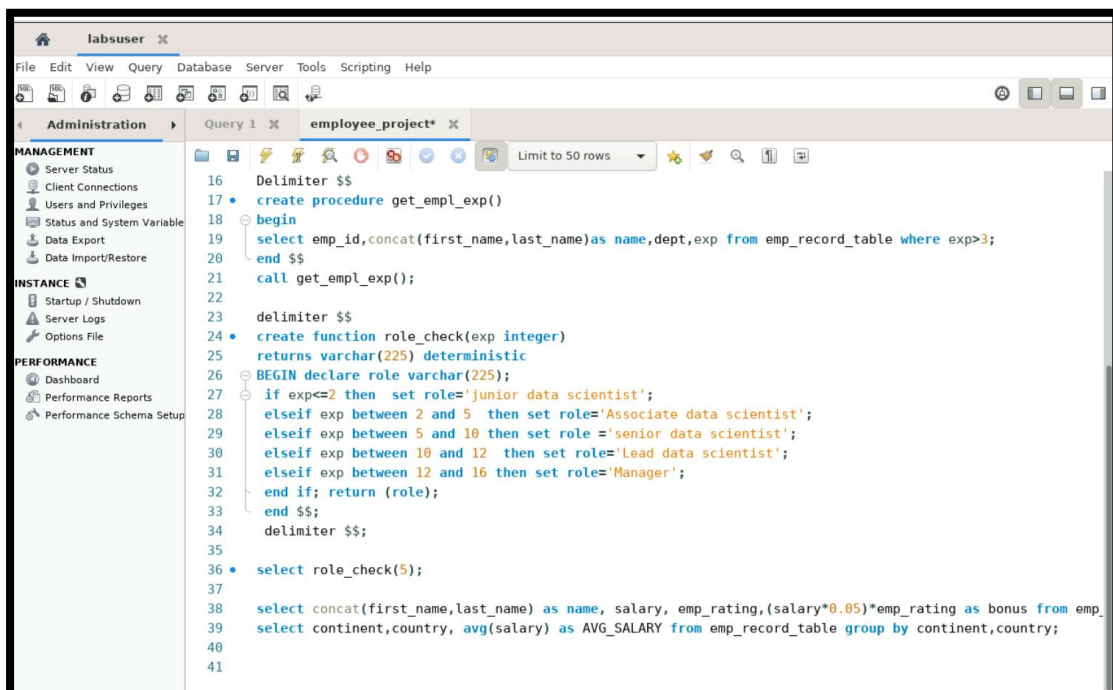
## Query for above statements:



The screenshot shows the SQL Developer interface with a query window titled 'employee\_project\*'. The query script is as follows:

```
1 • SELECT * FROM Employee.emp_record_table;
2 • use Employee;
3 • SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT,EMP_RATING from emp_record_table where EMP_RATING < 2;
4 • SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT,EMP_RATING from emp_record_table where EMP_RATING >4;
5 • SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT,EMP_RATING from emp_record_table where EMP_RATING between 2 and 4;
6 • select concat(first_name,last_name) as Name,dept from emp_record_table where dept = 'FINANCE';
7 • select emp_id,first_name,last_name,role from emp_record_table where emp_id in (select distinct (E.manager_id)
8 • select emp_id,first_name,last_name,dept from emp_record_table where dept = 'Finance' union
9 • select E.emp_id,E.first_name,E.last_name,E.dept from emp_record_table E where dept = 'Healthcare';
10 • select emp_id,concat(first_name,last_name)as NAME,role,dept,exp,rank() over(order by exp DESC)as exp_ranking
11 from emp_record_table;
12 • select emp_id,first_name,last_name,dept,emp_rating,max(emp_rating) over(partition by dept)as Dept_max_rating
13 from emp_record_table;
14 • select role,min(salary)as Min_salary,max(salary) as Max_salary
15 from emp_record_table group by role;
16 Delimiter $$
17 • create procedure get_empl_exp()
18 • begin
19 • select emp_id,concat(first_name,last_name)as name,dept,exp from emp_record_table where exp>3;
20 • end $$
21 • call get_empl_exp();
22
23 delimiter $$
24 • create function role_check(exp integer)
25 returns varchar(225) deterministic
26 • BEGIN declare role varchar(225);
27 • if exp<=2 then set role='junior data scientist';
```

Query Completed



The screenshot shows the SQL Developer interface with a query window titled 'employee\_project\*'. The query script is as follows:

```
16 Delimiter $$
17 • create procedure get_empl_exp()
18 • begin
19 • select emp_id,concat(first_name,last_name)as name,dept,exp from emp_record_table where exp>3;
20 • end $$
21 • call get_empl_exp();
22
23 delimiter $$
24 • create function role_check(exp integer)
25 returns varchar(225) deterministic
26 • BEGIN declare role varchar(225);
27 • if exp<=2 then set role='junior data scientist';
28 • elseif exp between 2 and 5 then set role='Associate data scientist';
29 • elseif exp between 5 and 10 then set role='senior data scientist';
30 • elseif exp between 10 and 12 then set role='Lead data scientist';
31 • elseif exp between 12 and 16 then set role='Manager';
32 • end if; return (role);
33 • end $$;
34 delimiter $$;
35
36 • select role_check(5);
37
38 • select concat(first_name,last_name) as name, salary, emp_rating,(salary*0.05)*emp_rating as bonus from emp_
39 • select continent,country, avg(salary) as AVG_SALARY from emp_record_table group by continent,country;
40
41
```