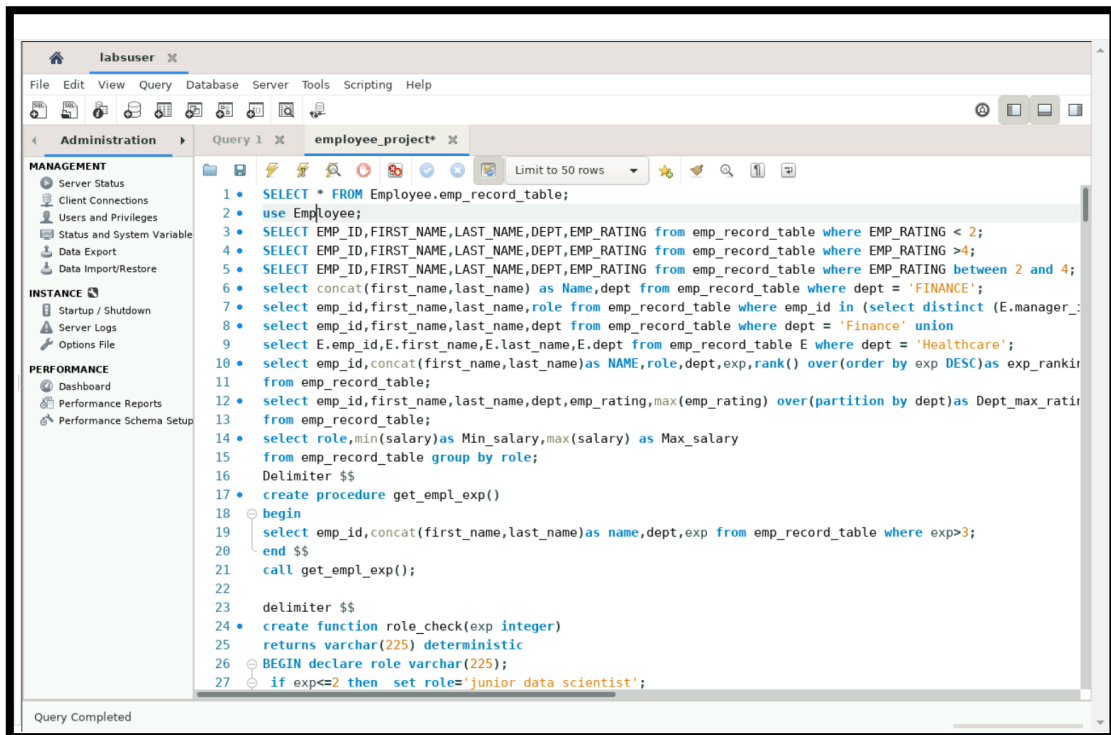


ScienceQtech Employee Performance Mapping

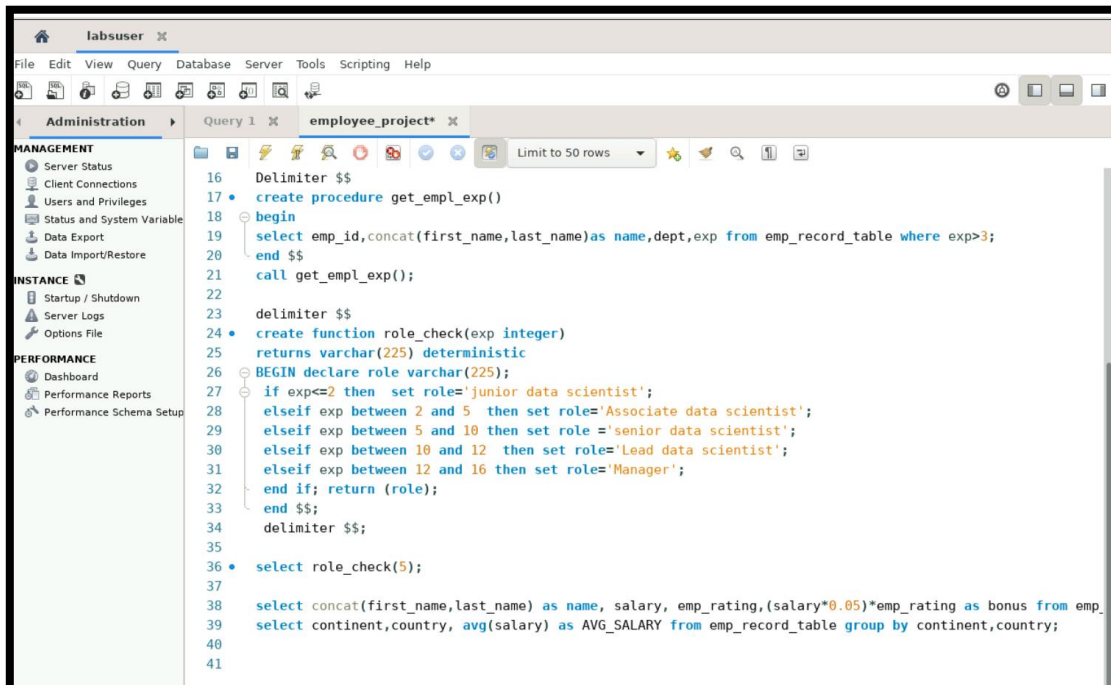
MySQL – Project Source Code



The screenshot shows the MySQL Workbench interface with the 'employee_project' database selected. The SQL editor contains the following code:

```
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 continuation of the SQL script in the MySQL Workbench editor:

```
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
```