**ScienceQtech Employee Performance Mapping: Course-end Project 1**

**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 on the project that the employees have undertaken, to analyze the employee database and extract specific data based on different requirements.

**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. As a DBA, 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 extra cost for expenses. This will raise the overall performance of the organization by ensuring that all required employees receive training.

**Note:** You must download the dataset from the course resource section in LMS and create a table to perform the above objective.

**Dataset description:**

**emp\_record\_table:** It contains the information of all the employees.

EMP\_ID – ID of the employee

FIRST\_NAME – First name of the employee

LAST\_NAME – Last name of the employee

GENDER – Gender of the employee

ROLE – Post of the employee

DEPT – Field of the employee

EXP – Years of experience the employee has

COUNTRY – Country in which the employee is presently living

CONTINENT – Continent in which the country is

SALARY – Salary of the employee

EMP\_RATING – Performance rating of the employee

MANAGER\_ID – The manager under which the employee is assigned

PROJ\_ID – The project on which the employee is working or has worked on

**Proj\_table:** It contains information about the projects.

PROJECT\_ID – ID for the project

PROJ\_Name – Name of the project

DOMAIN – Field of the project

START\_DATE – Day the project began

CLOSURE\_DATE – Day the project was or will be completed

DEV\_QTR – Quarter in which the project was scheduled

STATUS – Status of the project currently

**Data\_science\_team:** It contains information about all the employees in the Data Science team.

EMP\_ID – ID of the employee

FIRST\_NAME – First name of the employee

LAST\_NAME – Last name of the employee

GENDER – Gender of the employee

ROLE – Post of the employee

DEPT – Field of the employee

EXP – Years of experience the employee has

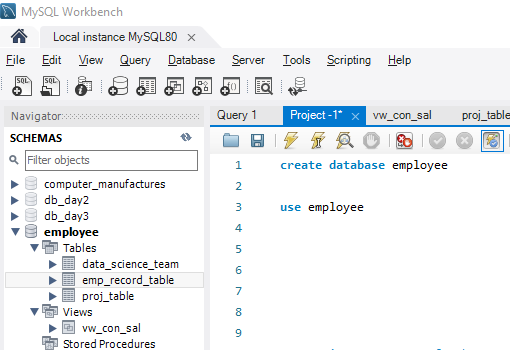
COUNTRY – Country in which the employee is presently living

CONTINENT – Continent in which the country is

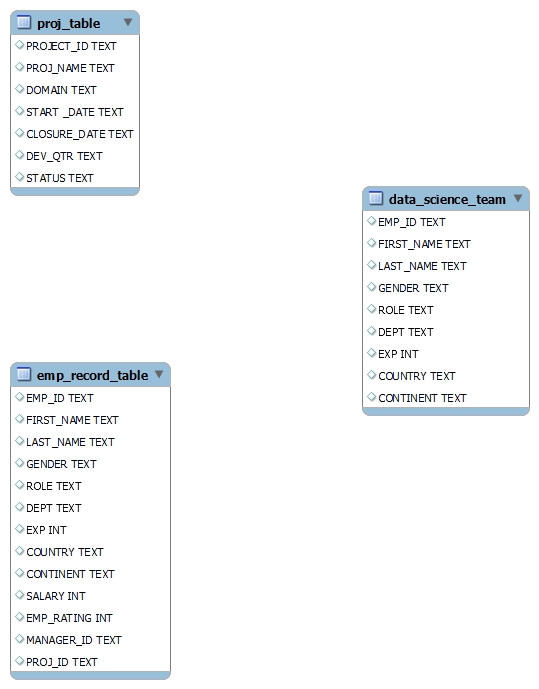
**The task to be performed:**

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.

**Query: create database employee**

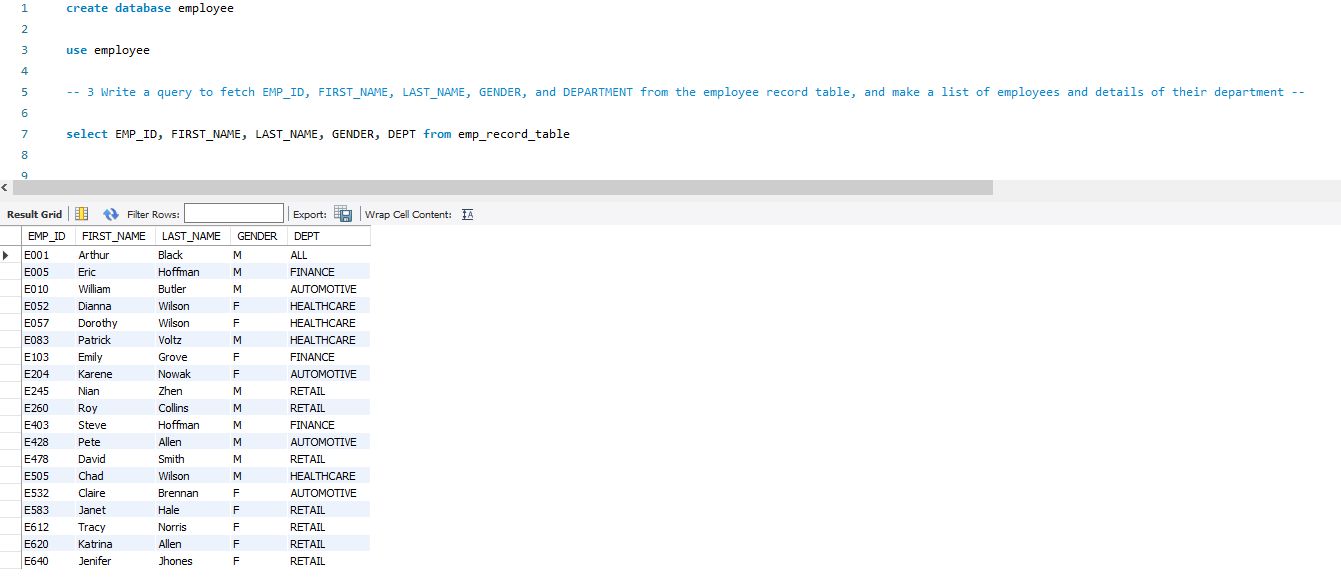


1. Create an ER diagram for the given employee database.



1. 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.

**Query: select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT from emp\_record\_table**

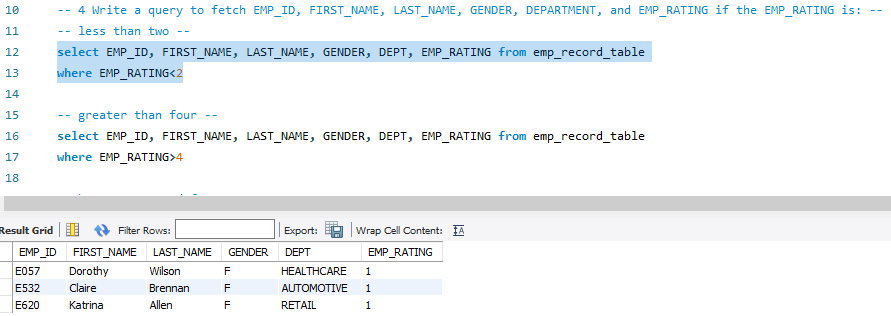
****

1. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:

* less than two

**Query: select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING from emp\_record\_table**

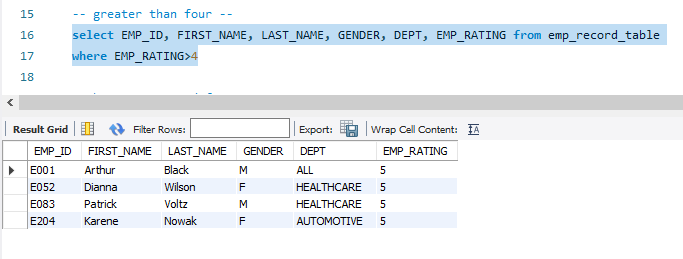
**where EMP\_RATING<2**



* greater than four

**Query: select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING from emp\_record\_table**

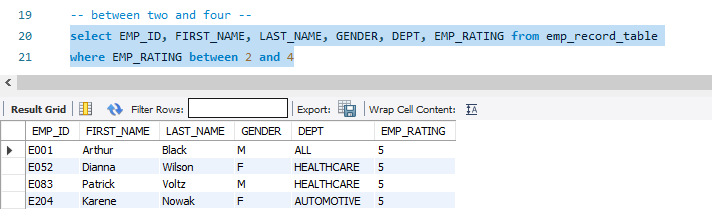
**where EMP\_RATING>4**



* between two and four

**Query: select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING from emp\_record\_table**

**where EMP\_RATING between 2 and 4**

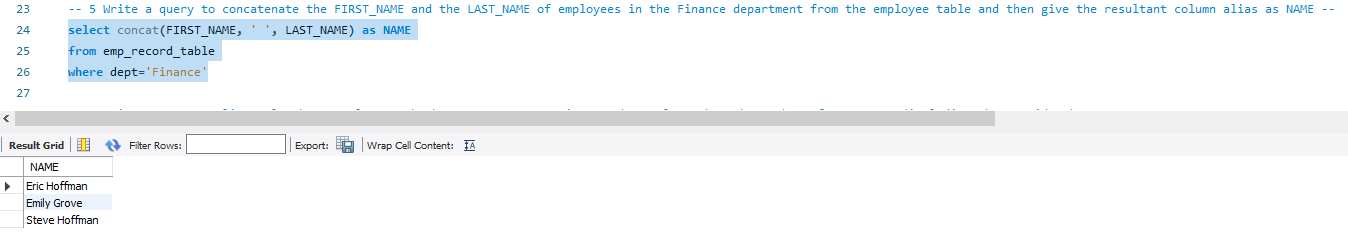


1. Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

**Query: select concat(FIRST\_NAME, ' ', LAST\_NAME) as NAME**

**from emp\_record\_table**

**where dept='Finance'**

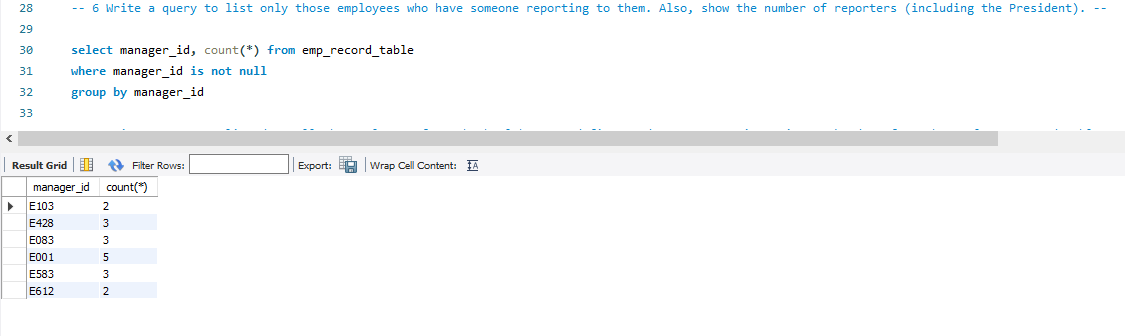


1. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).

**Query: select manager\_id, count(\*) from emp\_record\_table**

**where manager\_id is not null**

**group by manager\_id**



1. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.

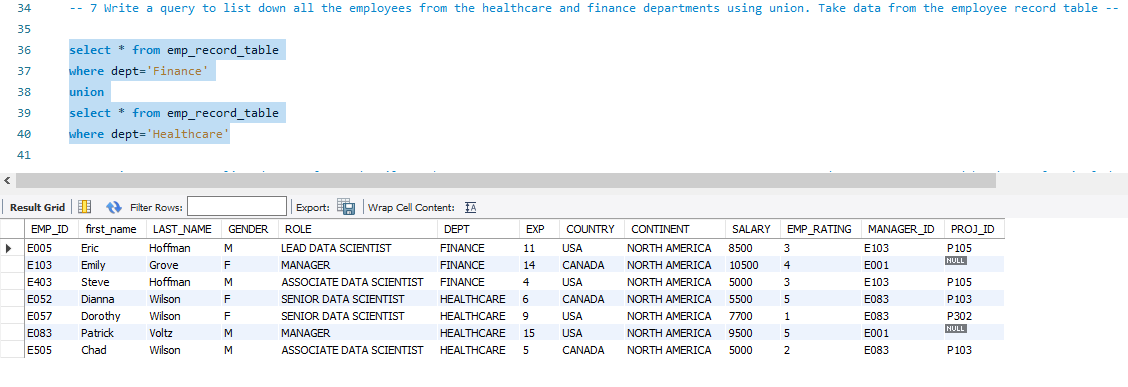
**Query: select \* from emp\_record\_table**

**where dept='Finance'**

**union**

**select \* from emp\_record\_table**

**where dept='Healthcare'**



1. 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.

**Query: select \* from (**

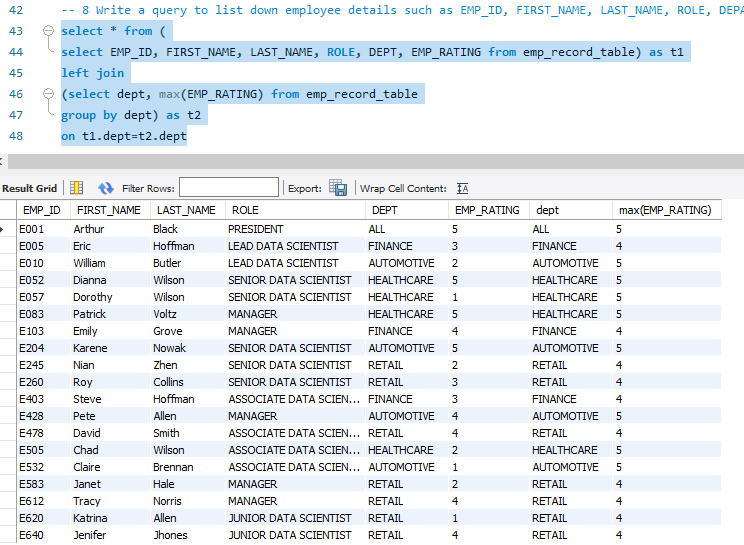
**select EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, EMP\_RATING from emp\_record\_table) as t1**

**left join**

**(select dept, max(EMP\_RATING) from emp\_record\_table**

**group by dept) as t2**

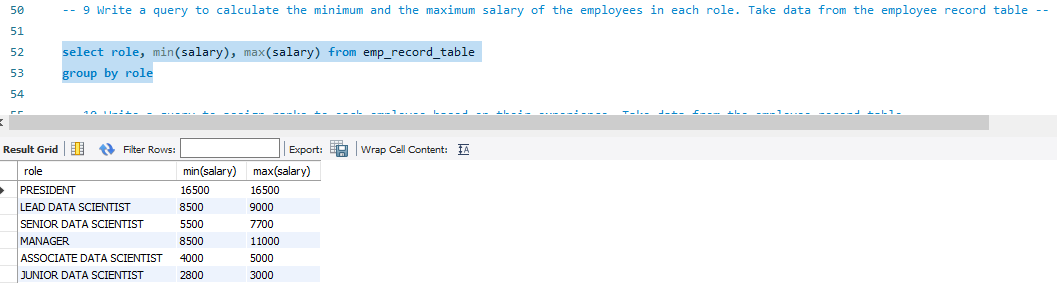
**on t1.dept=t2.dept**



1. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

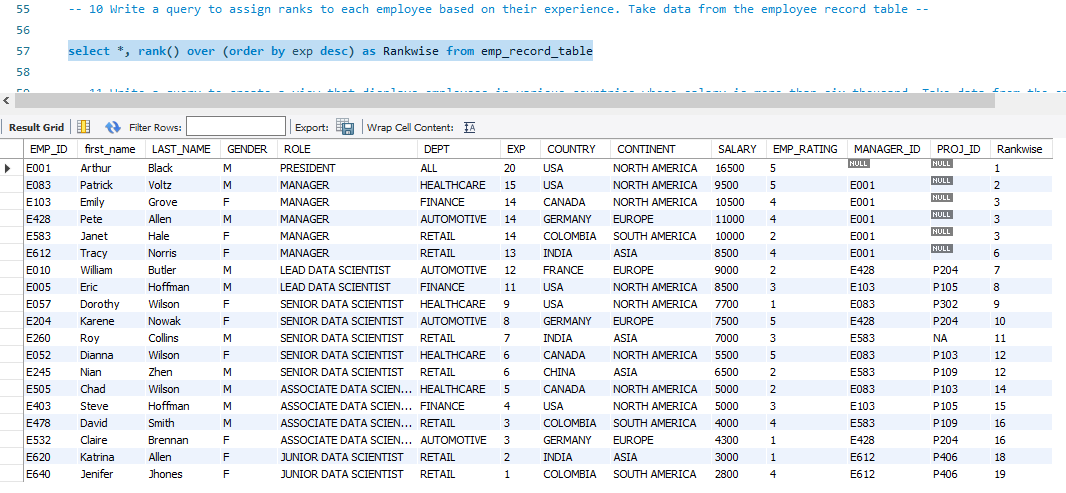
**Query: select role, min(salary), max(salary) from emp\_record\_table**

**group by role**



1. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

**Query: select \*, rank() over (order by exp desc) as Rankwise from emp\_record\_table**



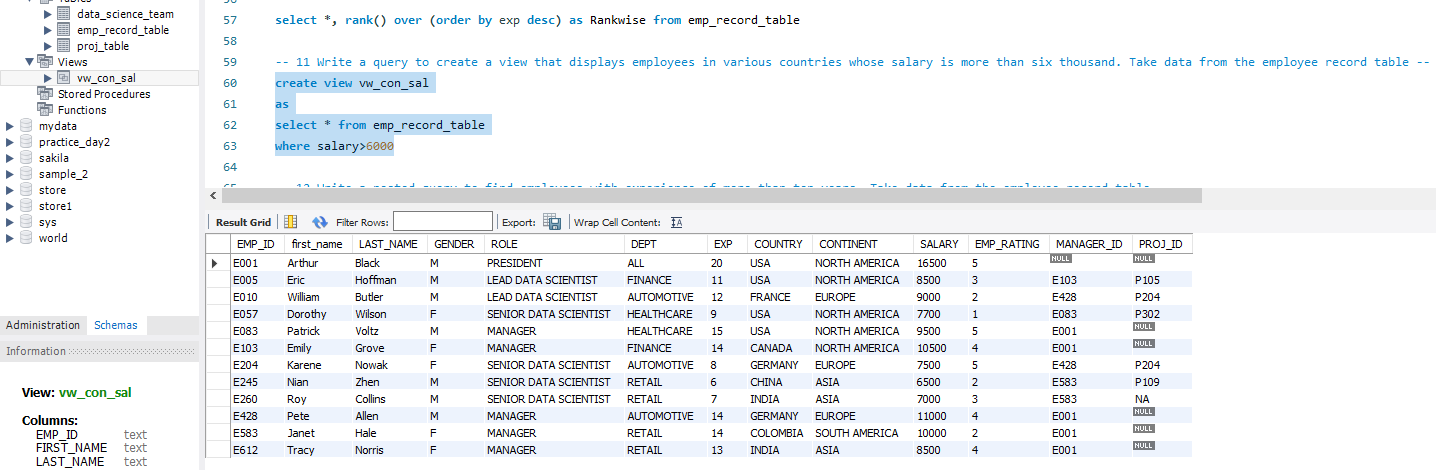
1. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.

**Query: create view vw\_con\_sal**

**as**

**select \* from emp\_record\_table**

**where salary>6000**

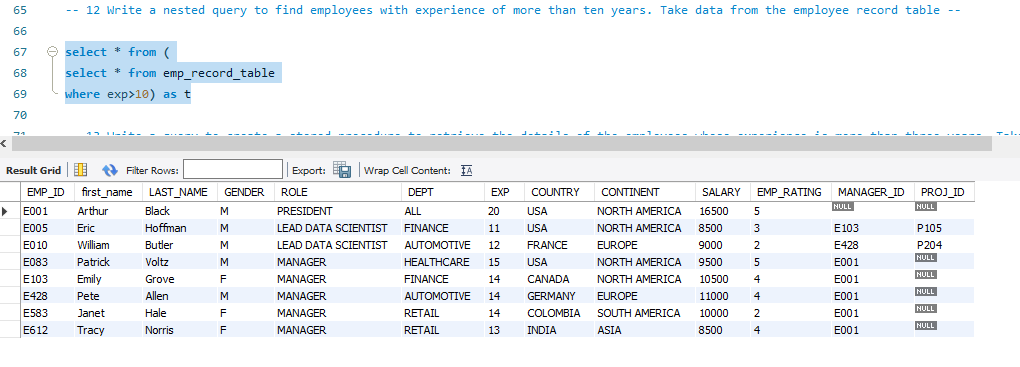


1. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

**Query: select \* from (**

**select \* from emp\_record\_table**

**where exp>10) as t**



1. 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.

**Query: delimiter //**

**create procedure sp\_exp\_three()**

**begin**

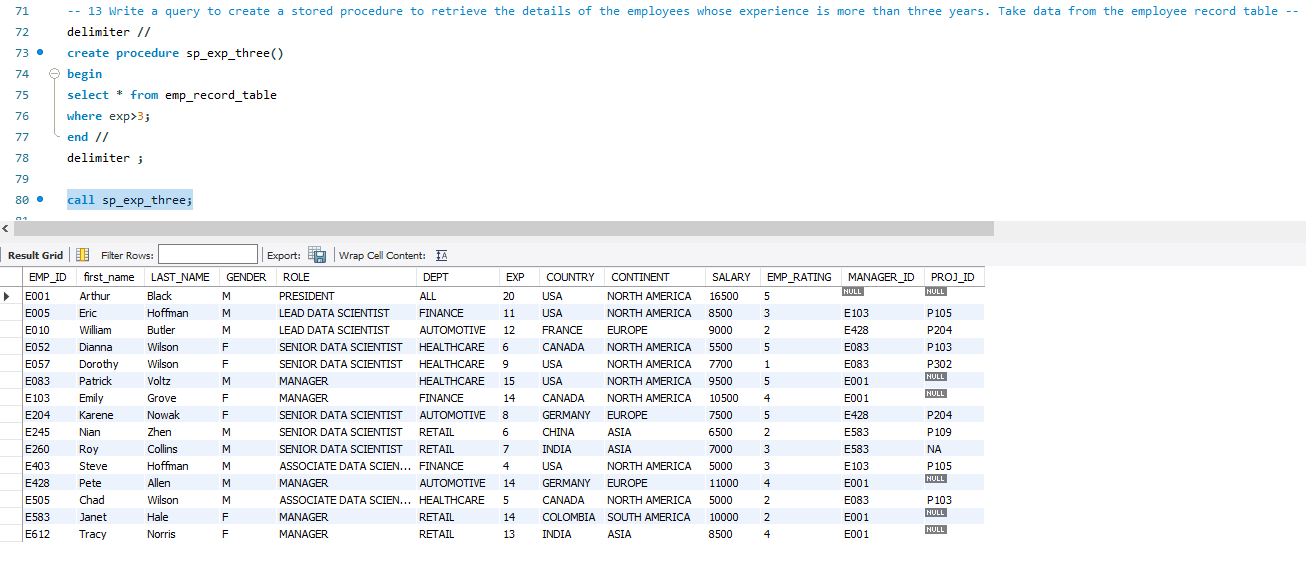
**select \* from emp\_record\_table**

**where exp>3;**

**end //**

**delimiter ;**

**call sp\_exp\_three;**



1. 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 being:

* 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'
* For an employee with the experience of 12 to 16 years assign 'MANAGER'

**Query: select \* , if(exp<=2, 'JUNIOR DATA SCIENTIST',**

**if(exp<=5, 'ASSOCIATE DATA SCIENTIST',**

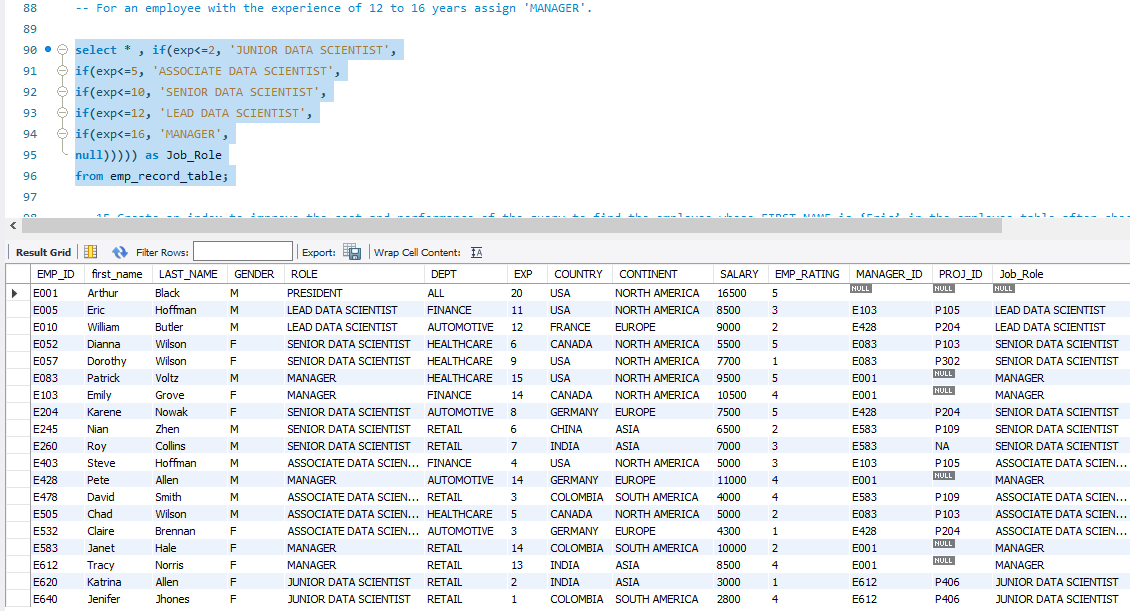
**if(exp<=10, 'SENIOR DATA SCIENTIST',**

**if(exp<=12, 'LEAD DATA SCIENTIST',**

**if(exp<=16, 'MANAGER',**

**null))))) as Job\_Role**

**from emp\_record\_table;**



1. 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.

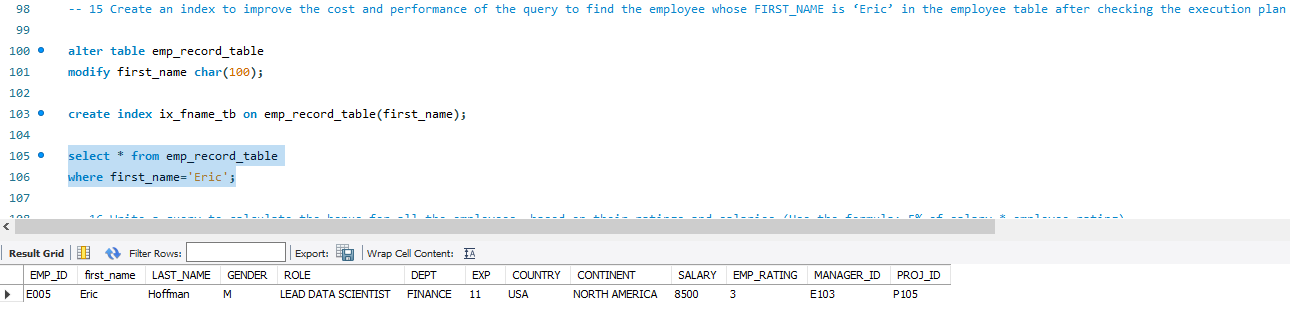
**Query: alter table emp\_record\_table**

**modify first\_name char(100);**

**create index ix\_fname\_tb on emp\_record\_table(first\_name);**

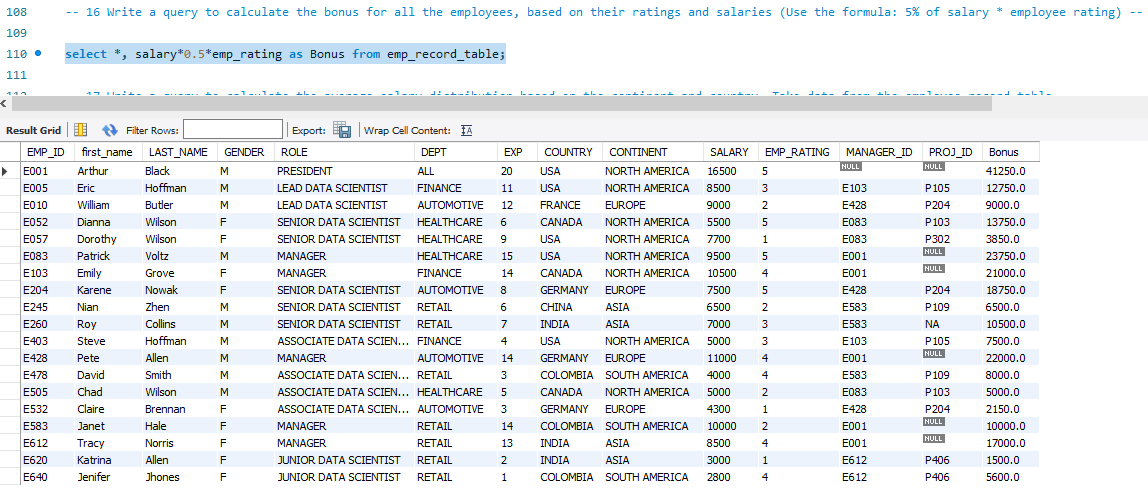
**select \* from emp\_record\_table**

**where first\_name='Eric';**



1. 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).

**Query: select \*, salary\*0.5\*emp\_rating as Bonus from emp\_record\_table;**



1. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.

**Query: select continent, country, avg(salary) from emp\_record\_table**

**group by continent, country**

