**ScienceQtech Employee Performance Mapping**

Filed by

JAGADEESWARAN A B

(Business Analyst)

Simplilearn SQL project I

**DESCRIPTION**

ScienceQtech is a start up 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 analyse 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.

**TOOLS USED**

MySQL , draw.io (ER diagram)

**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: (to create database)**

create database employee;

use employee;

**Action performed:**

Import emp\_record\_table into employee database using “Table data import wizard” option

**Query: (to describe employee record table)**

desc emp\_record\_table;

**Output:**



**Query: (to create and describe project table)**

create table proj\_table

(

project\_id varchar(255) not null,

proj\_name varchar(255) not null,

domain varchar(255) not null,

start\_date date,

closure\_date date,

dev\_qtr varchar(255) not null,

proj\_status varchar(255) not null,

primary key(project\_id)

);

desc proj\_table;

**Output:**



**Query: (to create and describe data science team table)**

create table data\_science\_team

(

emp\_id varchar(255) not null,

first\_name varchar(255) not null,

last\_name varchar(255) not null,

gender varchar(255) not null,

emp\_role varchar(255) not null,

emp\_dept varchar(255) not null,

emp\_exp int(20) not null,

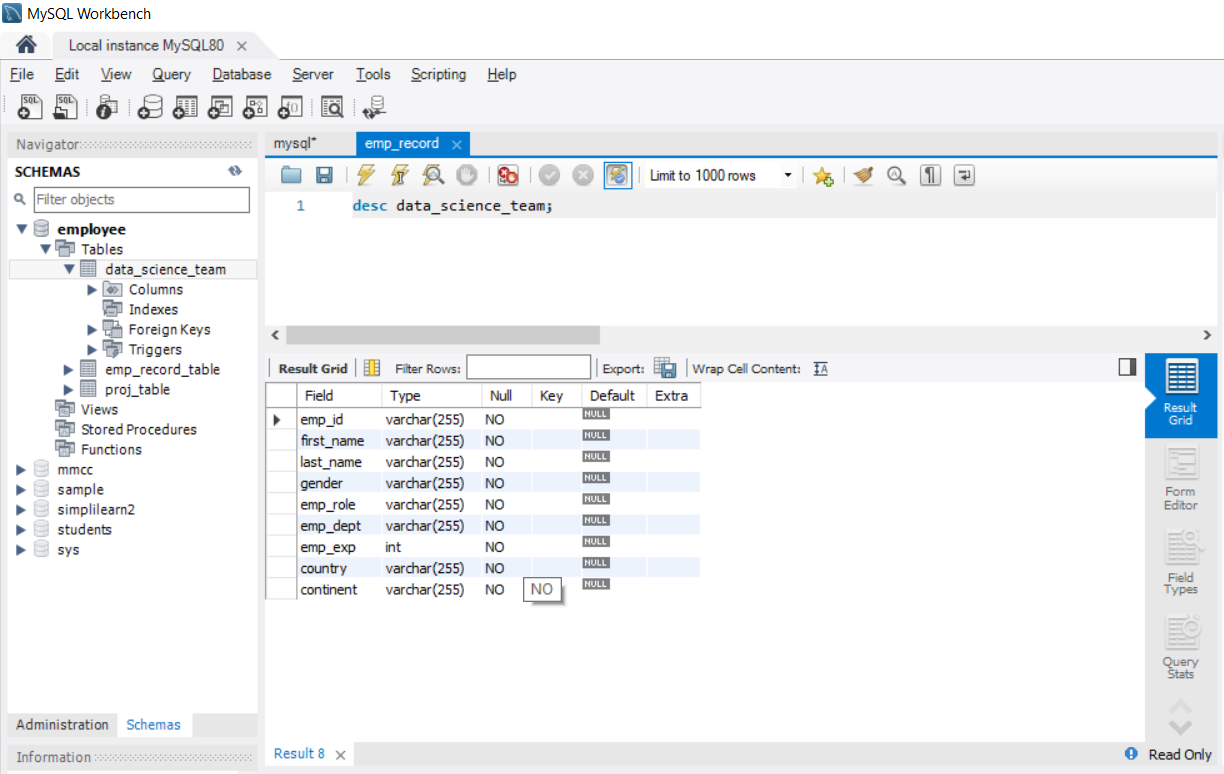
country varchar(255) not null,

continent varchar(255) not null

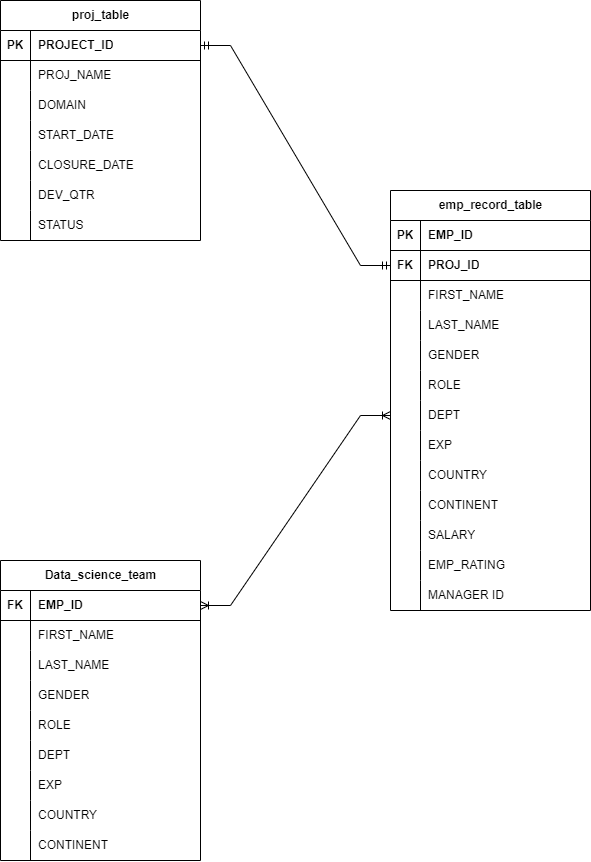
);

desc data\_science\_team;

**Output:**



**2. Create an ER diagram for the given employee database**.

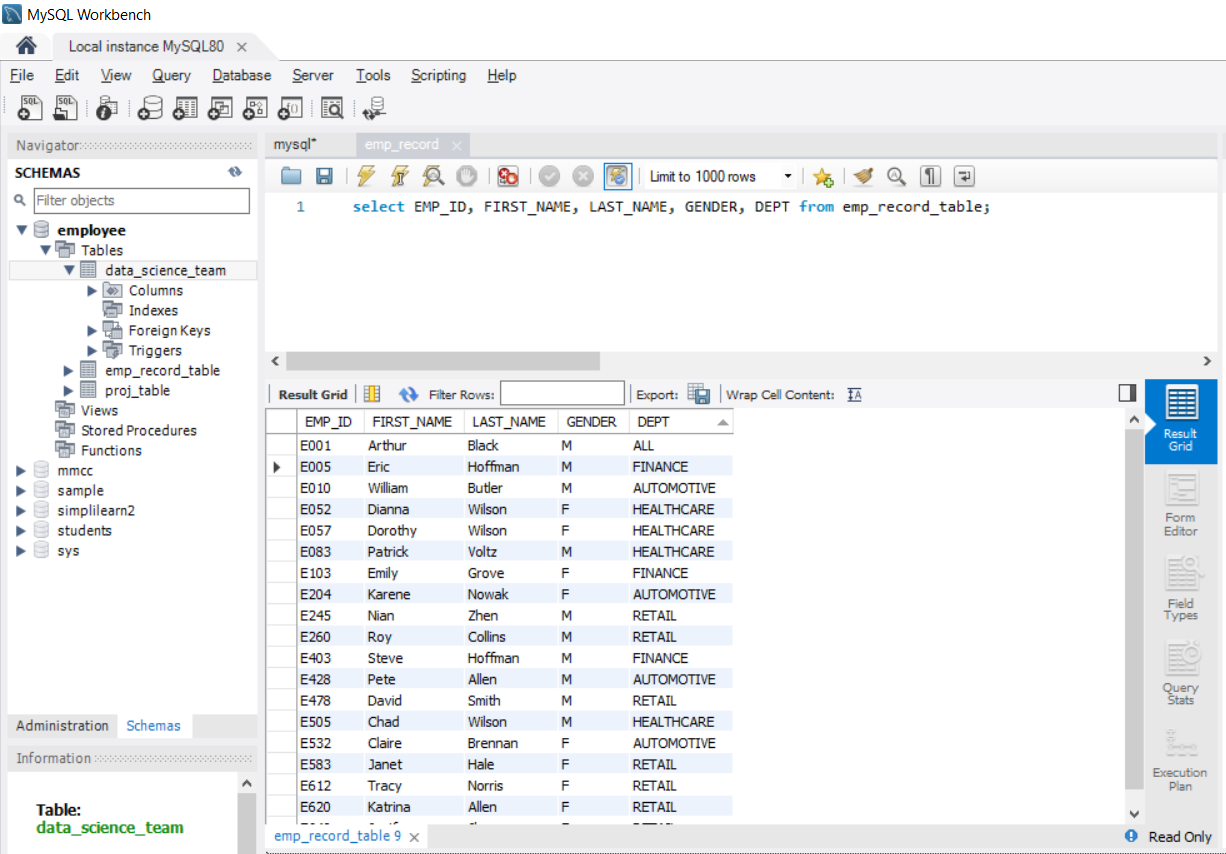


**3. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.**

**Query:**

Select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT from emp\_record\_table;

**Output:**



**4. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING:**

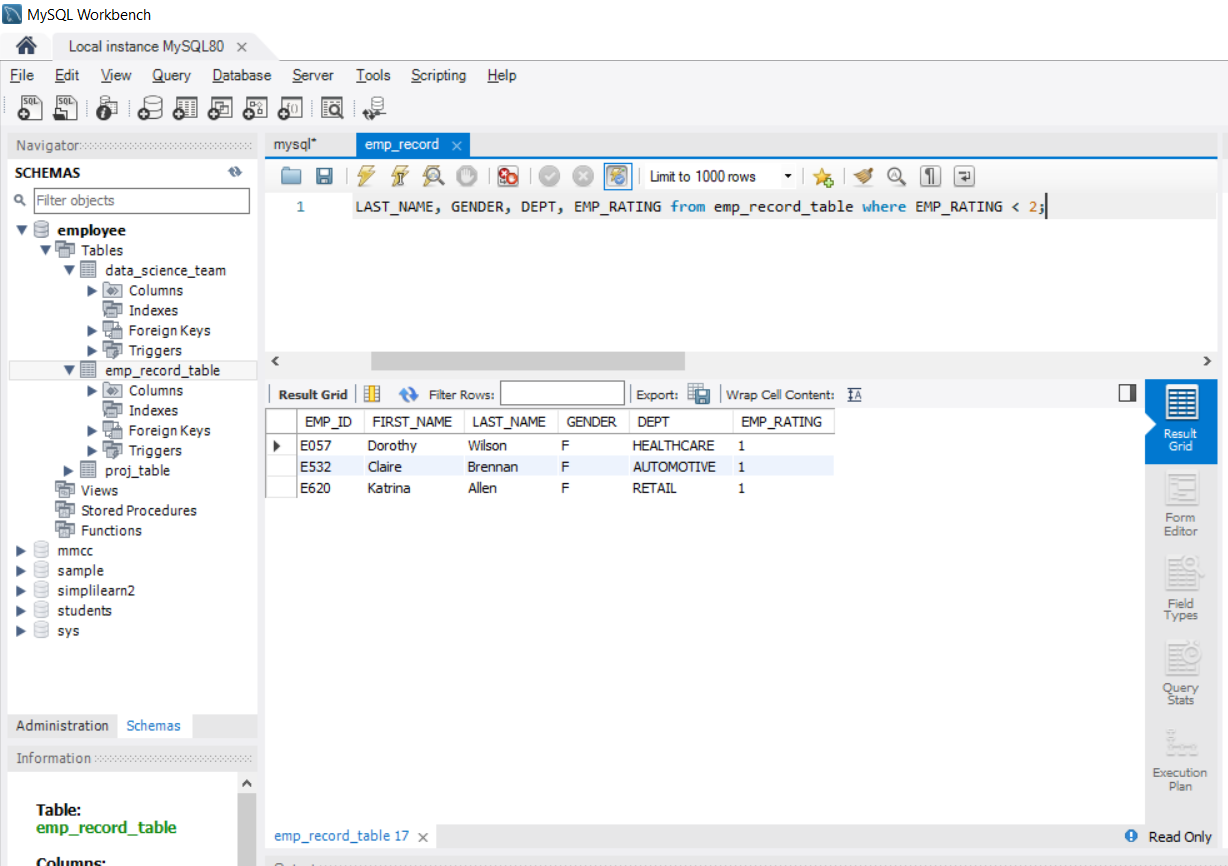
**Condition 1:**

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;

**Output:**



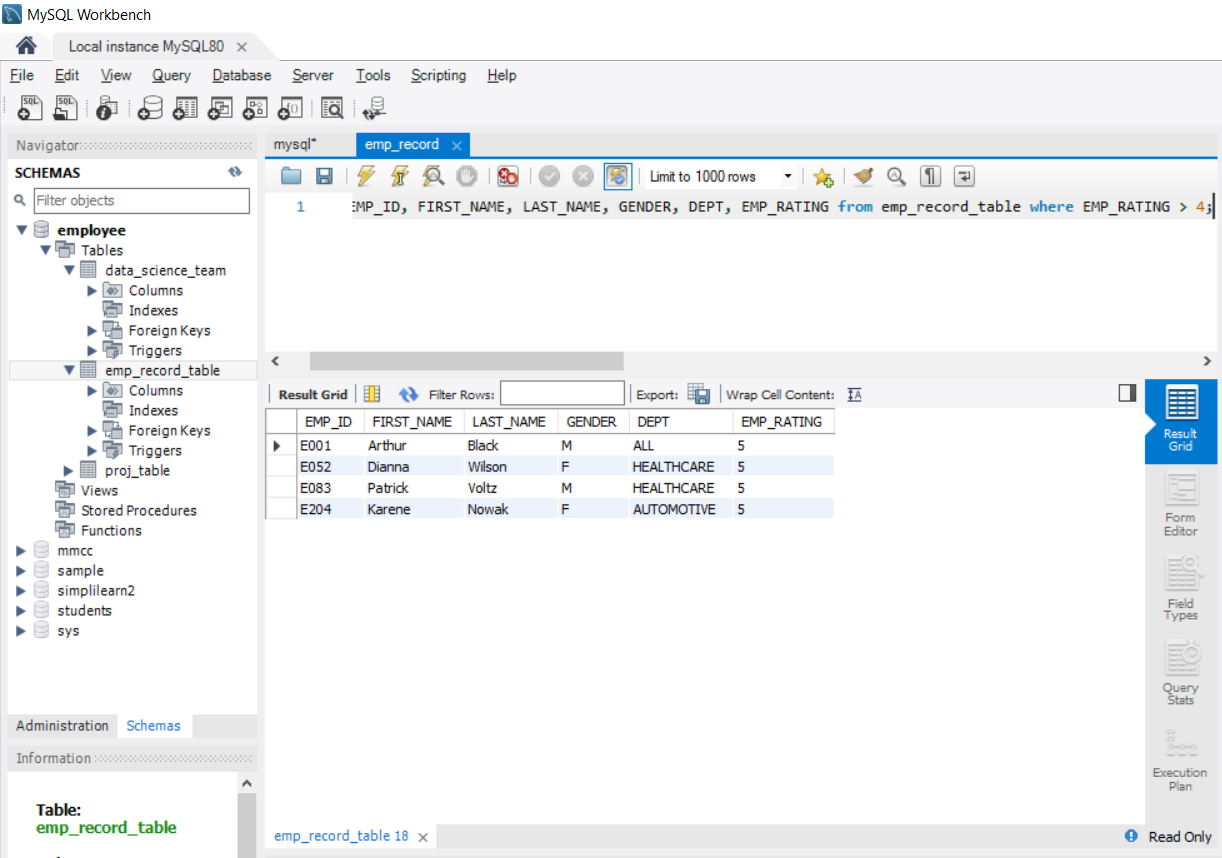
**Condition 2:**

If the EMP\_RATING is greater than 4

**Query:**

Select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING from emp\_record\_table where EMP\_RATING > 4;

**Output:**



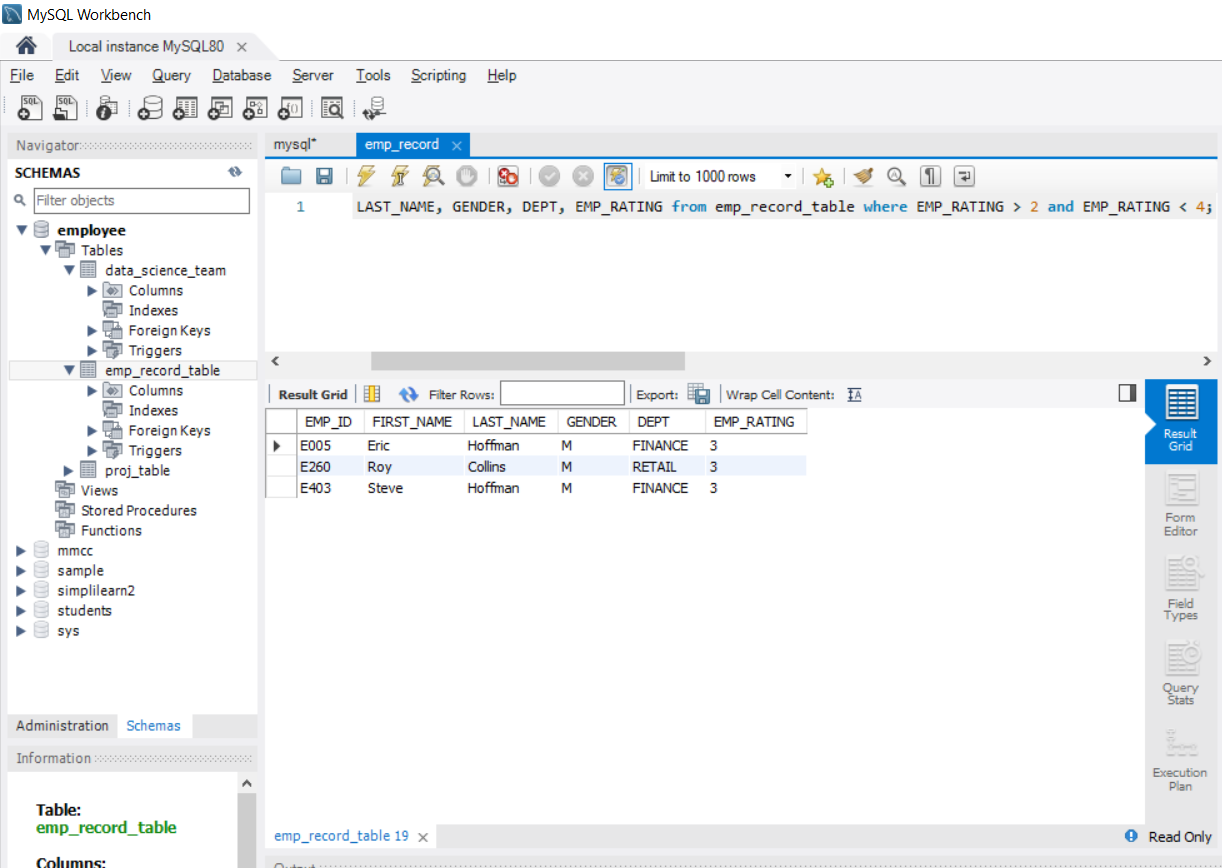
**Condition 3:**

If the EMP\_RATING is between two and four

**Query:**

Select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING from emp\_record\_table where EMP\_RATING > 2 and EMP\_RATING < 4;

**Output:**

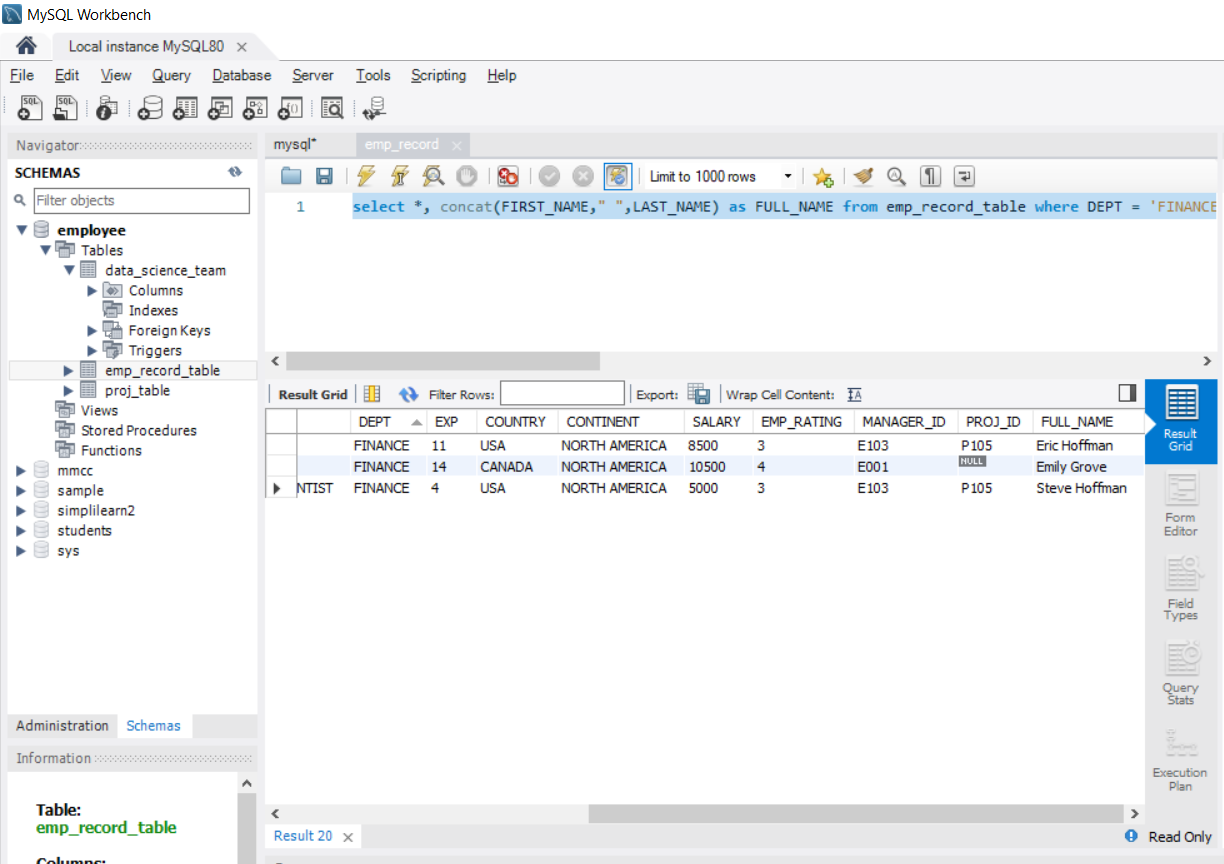


**5. Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.**

**Query:**

Select \*, concat(FIRST\_NAME," ",LAST\_NAME) as FULL\_NAME from emp\_record\_table where DEPT = 'FINANCE' ;

**Output:**

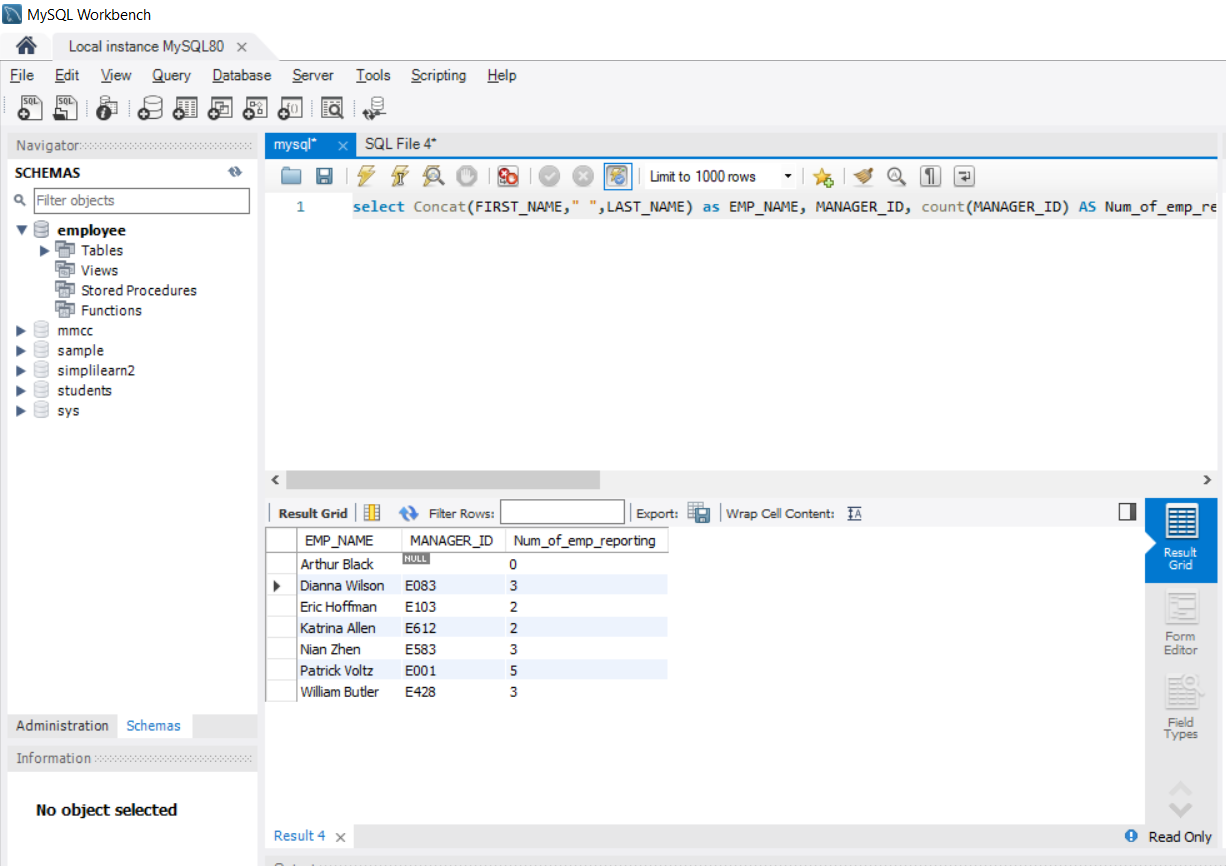


**6. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).**

**Query:**

Select Concat(FIRST\_NAME," ",LAST\_NAME) as EMP\_NAME, MANAGER\_ID, count(MANAGER\_ID) AS Num\_of\_emp\_reporting from emp\_record\_table GROUP BY MANAGER\_ID ORDER BY Num\_of\_emp\_reporting DESC;

**Output:**



**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.**

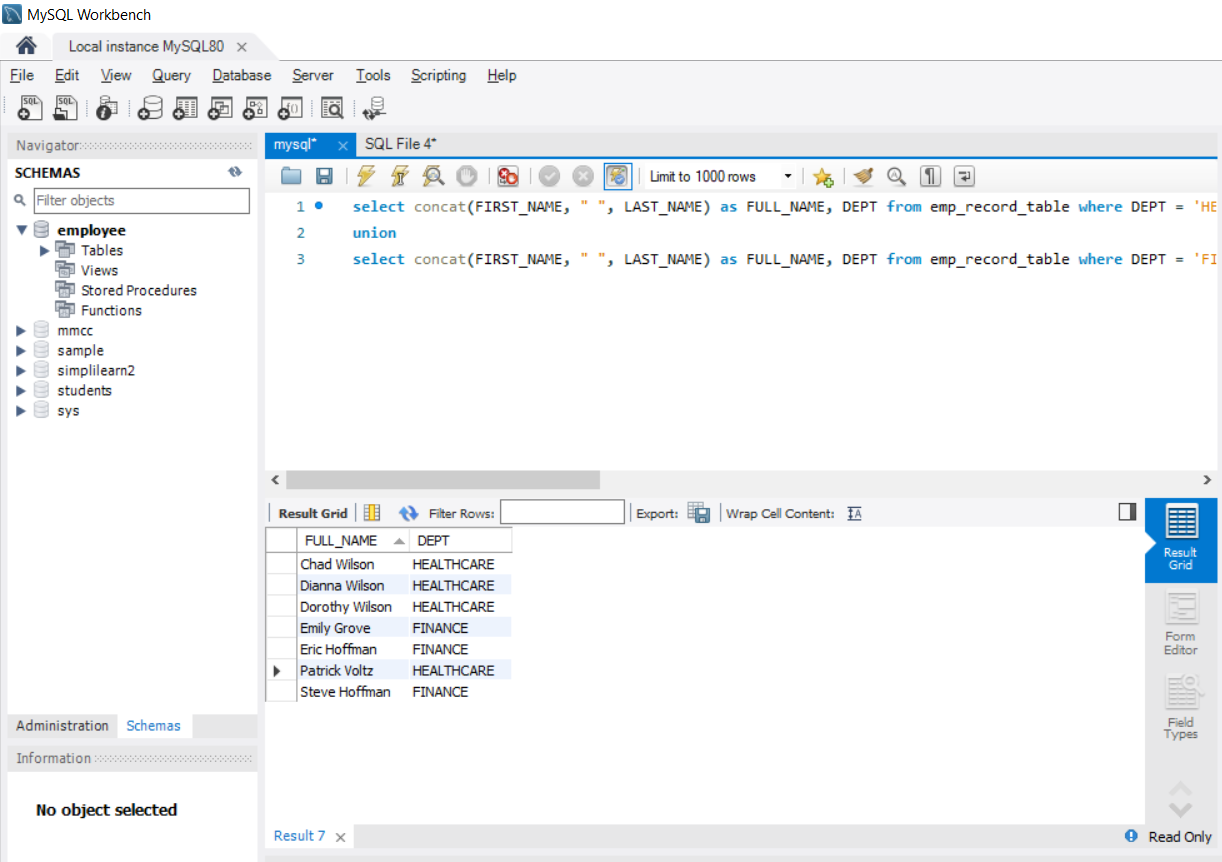
**Query:**

select concat(FIRST\_NAME, " ", LAST\_NAME) as FULL\_NAME, DEPT from emp\_record\_table where DEPT = 'HEALTHCARE'

union

select concat(FIRST\_NAME, " ", LAST\_NAME) as FULL\_NAME, DEPT from emp\_record\_table where DEPT = 'FINANCE';

**Output:**

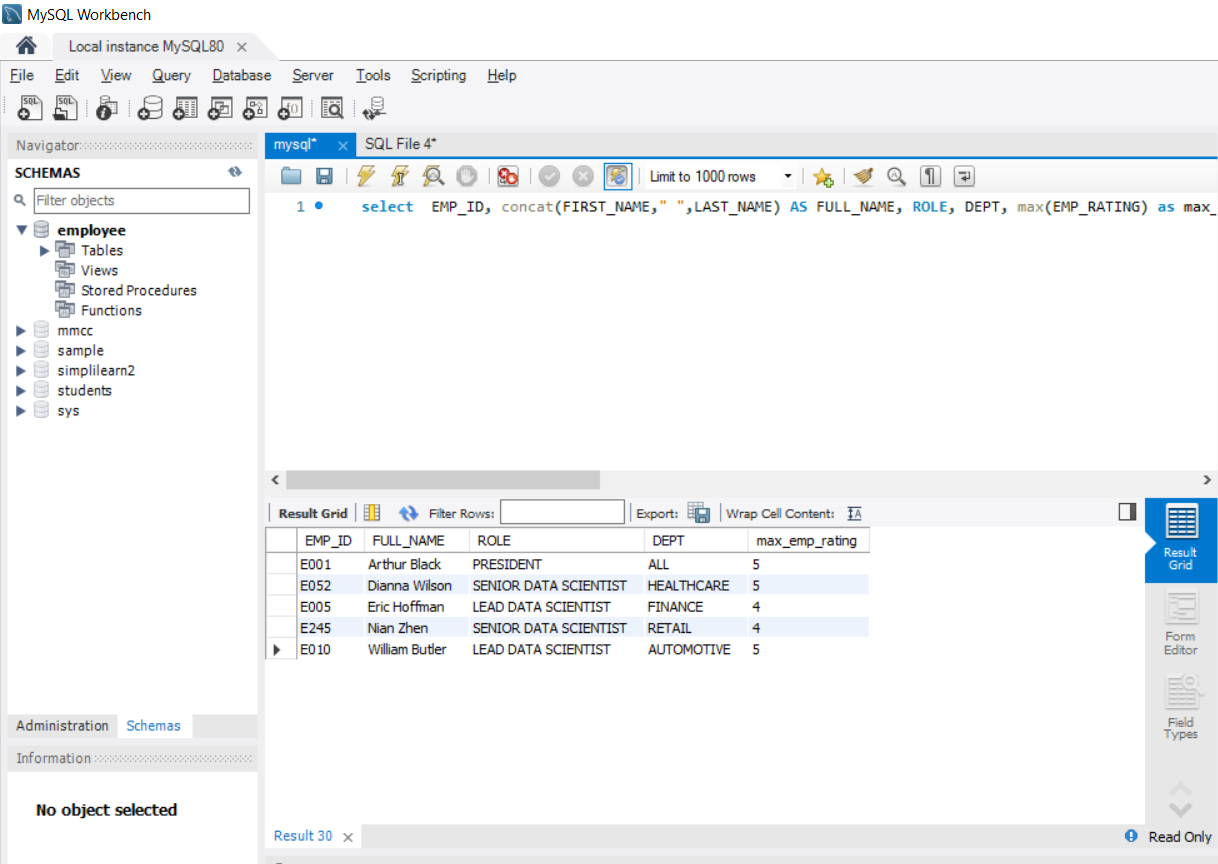


**8. Write a query to list down employee details such as EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPARTMENT, and EMP\_RATING grouped by dept. Also, include the respective employee rating along with the max emp rating for the department.**

**Query:**

select EMP\_ID, concat(FIRST\_NAME," ",LAST\_NAME) AS FULL\_NAME, ROLE, DEPT, max(EMP\_RATING) as max\_emp\_rating from (select EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, EMP\_RATING from emp\_record\_table order by DEPT) temptable group by DEPT;

**Output:**

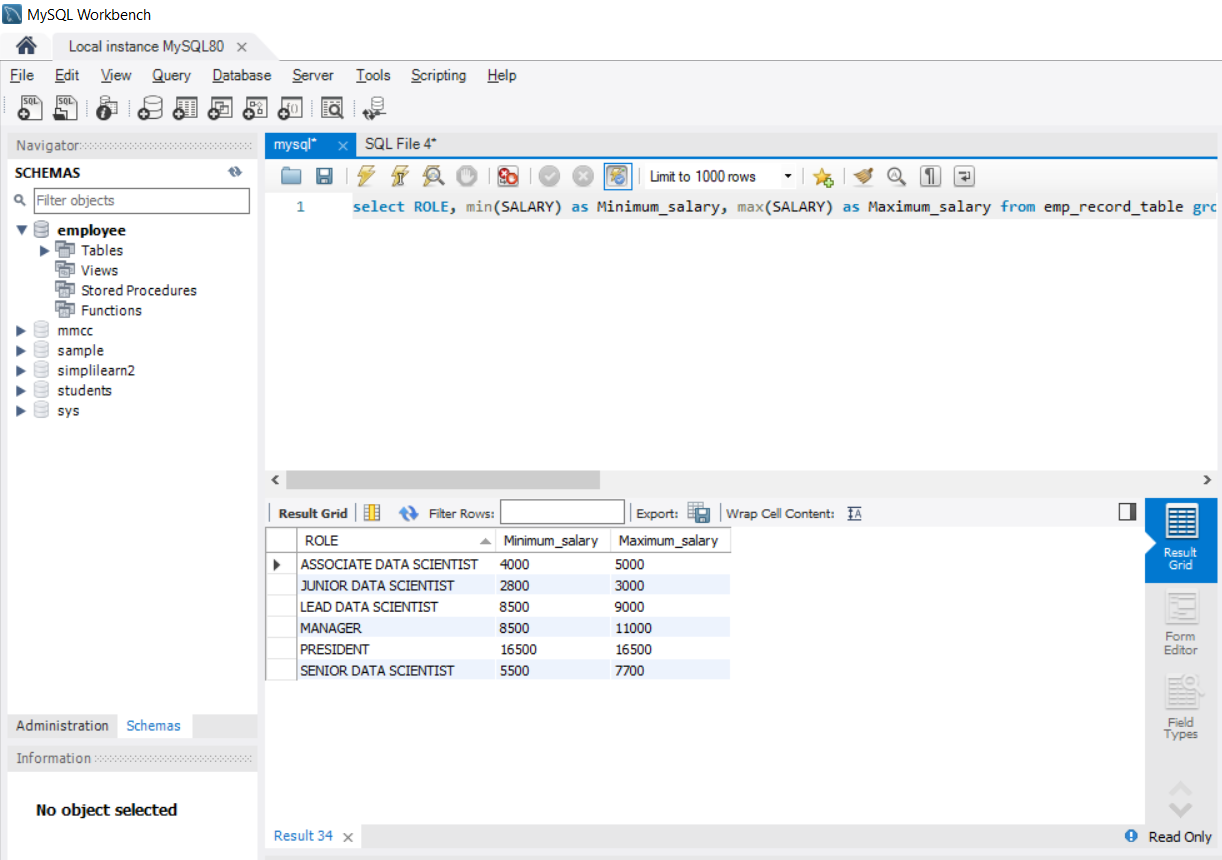


**9. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.**

**Query:**

select ROLE, min(SALARY) as Minimum\_salary, max(SALARY) as Maximum\_salary from emp\_record\_table group by ROLE;

**Output:**

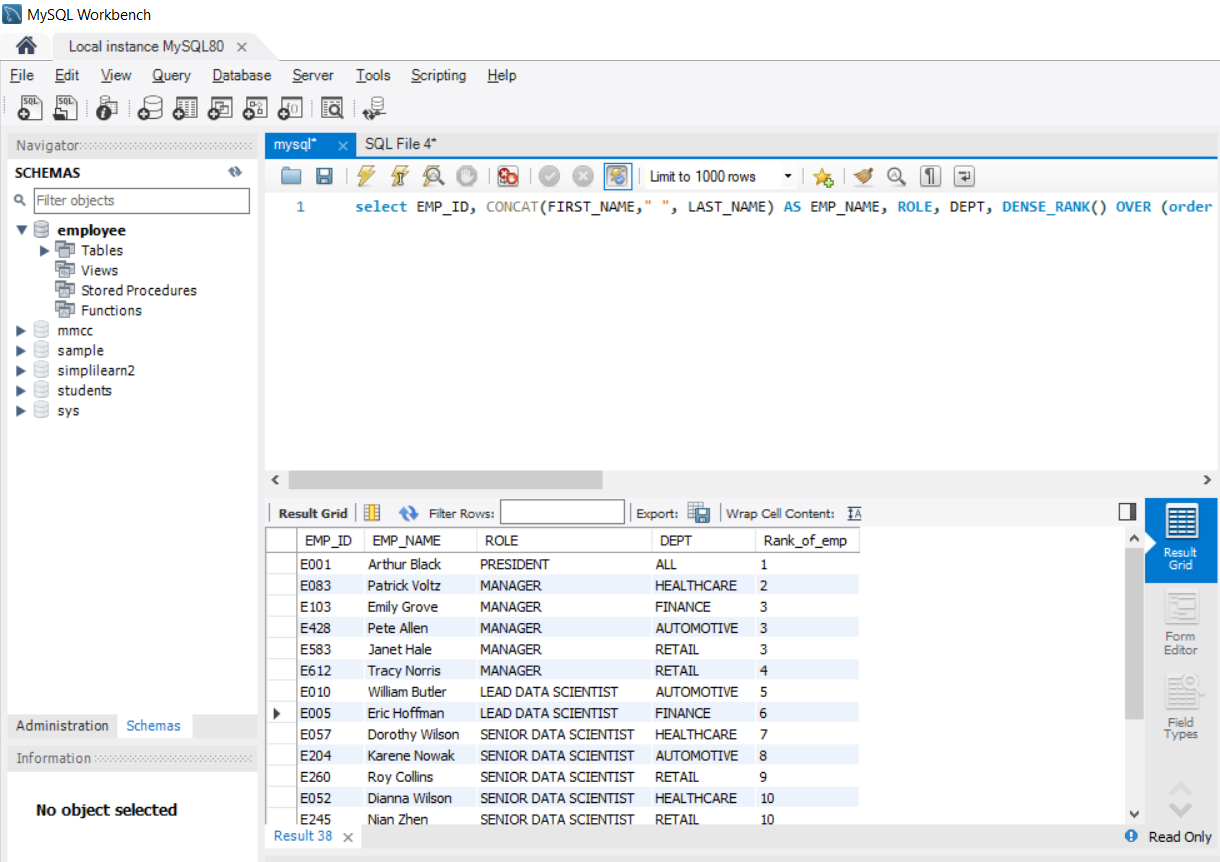


**10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.**

**Query:**

Select EMP\_ID, CONCAT(FIRST\_NAME," ", LAST\_NAME) AS EMP\_NAME, ROLE, DEPT, DENSE\_RANK() OVER (order by EXP desc) as Rank\_of\_emp from emp\_record\_table;

**Output:**



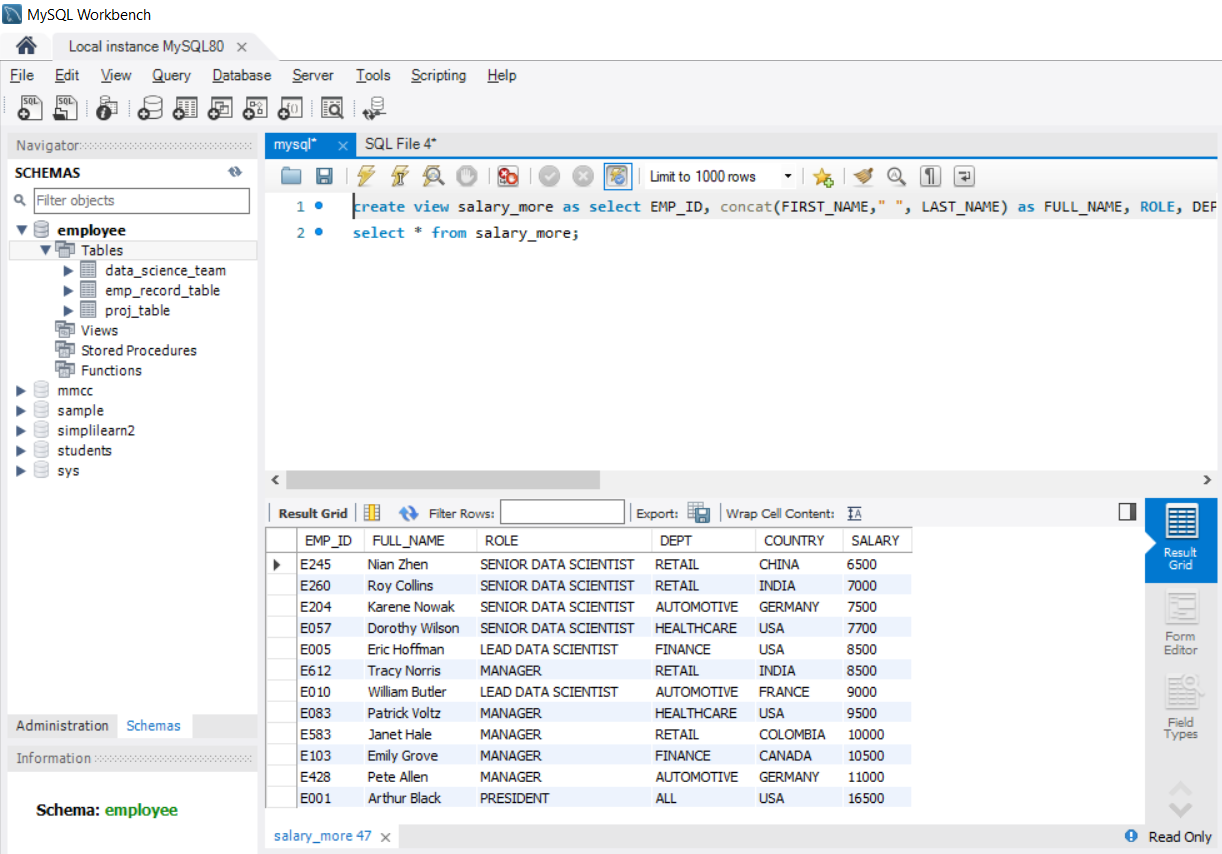
**11. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.**

**Query:**

Create view salary\_more as select EMP\_ID, concat(FIRST\_NAME," ", LAST\_NAME) as FULL\_NAME, ROLE, DEPT, COUNTRY, SALARY from emp\_record\_table where salary > 6000 order by SALARY;

select \* from salary\_more;

**Output:**

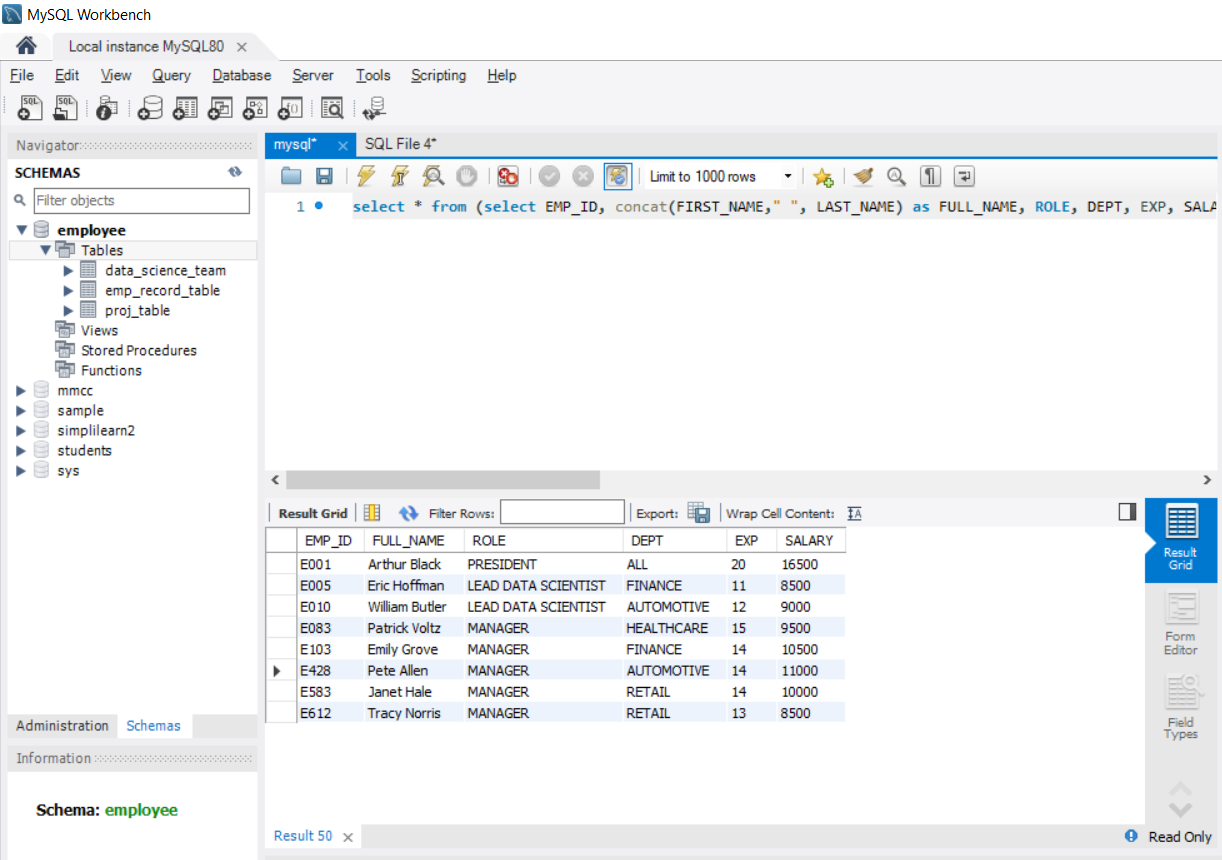


**12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.**

**Query:**

Select \* from (select EMP\_ID, concat(FIRST\_NAME," ", LAST\_NAME) as FULL\_NAME, ROLE, DEPT, EXP, SALARY from emp\_record\_table) temptable where EXP > 10 order by EXP desc;

**Output:**



**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.**

***STORED PROCEDURE:***

CREATE DEFINER=`root`@`localhost` PROCEDURE `Employee\_exp`(in EXP int)

BEGIN

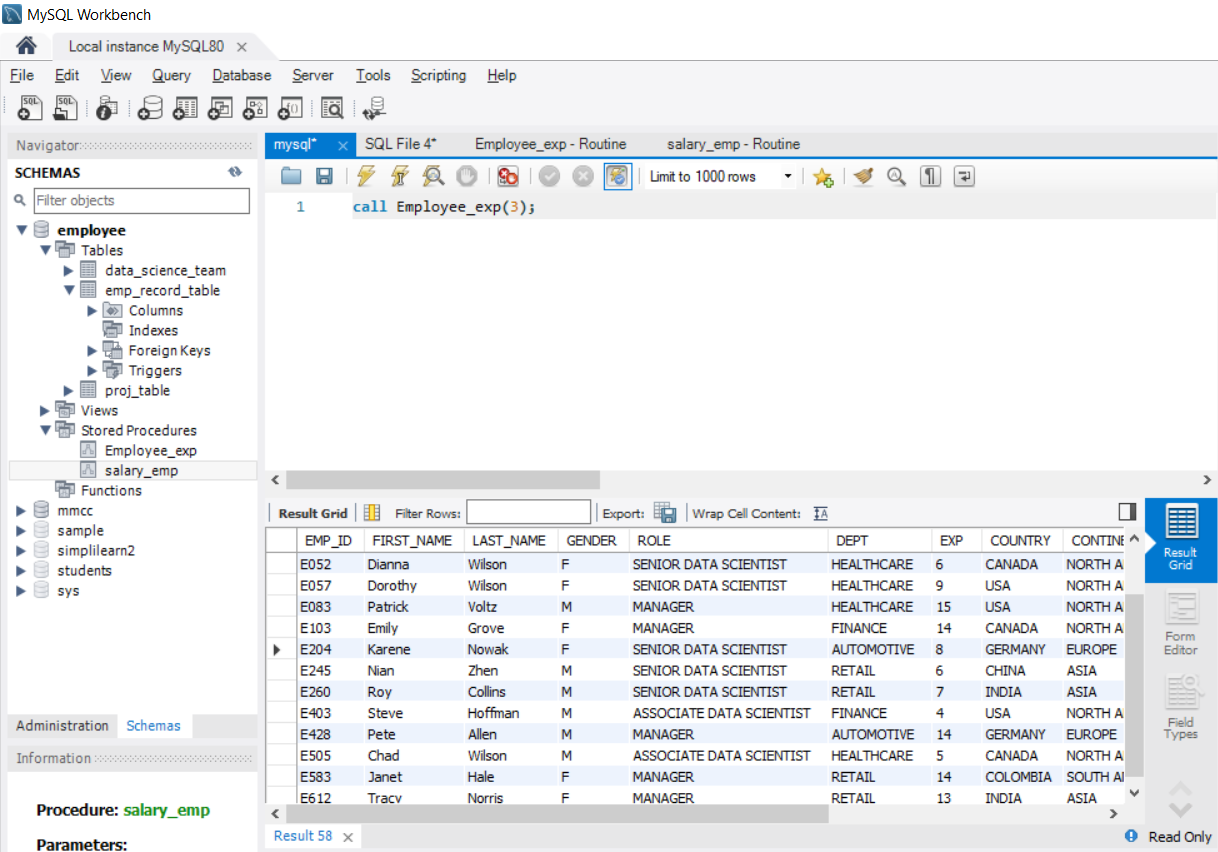
Select \* from emp\_record\_table where emp\_record\_table.EXP > EXP order by emp\_record\_table.EXP;

END

**Query:**

call Employee\_exp(3);

**Output:**



**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'

***STANDARD PROCEDURE:***

CREATE DEFINER=`root`@`localhost` PROCEDURE `Emp\_role`(OUT Employee\_ROLE varchar(255))

BEGIN

DECLARE Employee\_ROLE VARCHAR(40);

IF EXP>12 AND 16 THEN

SET Employee\_ROLE="MANAGER";

ELSEIF EXP>10 AND 12 THEN

SET Employee\_ROLE ="LEAD DATA SCIENTIST";

ELSEIF EXP>5 AND 10 THEN

SET Employee\_ROLE ="SENIOR DATA SCIENTIST";

ELSEIF EXP>2 AND 5 THEN

SET Employee\_ROLE ="ASSOCIATE DATA SCIENTIST";

ELSEIF EXP<=2 THEN

SET Employee\_ROLE ="JUNIOR DATA SCIENTIST";

END IF;

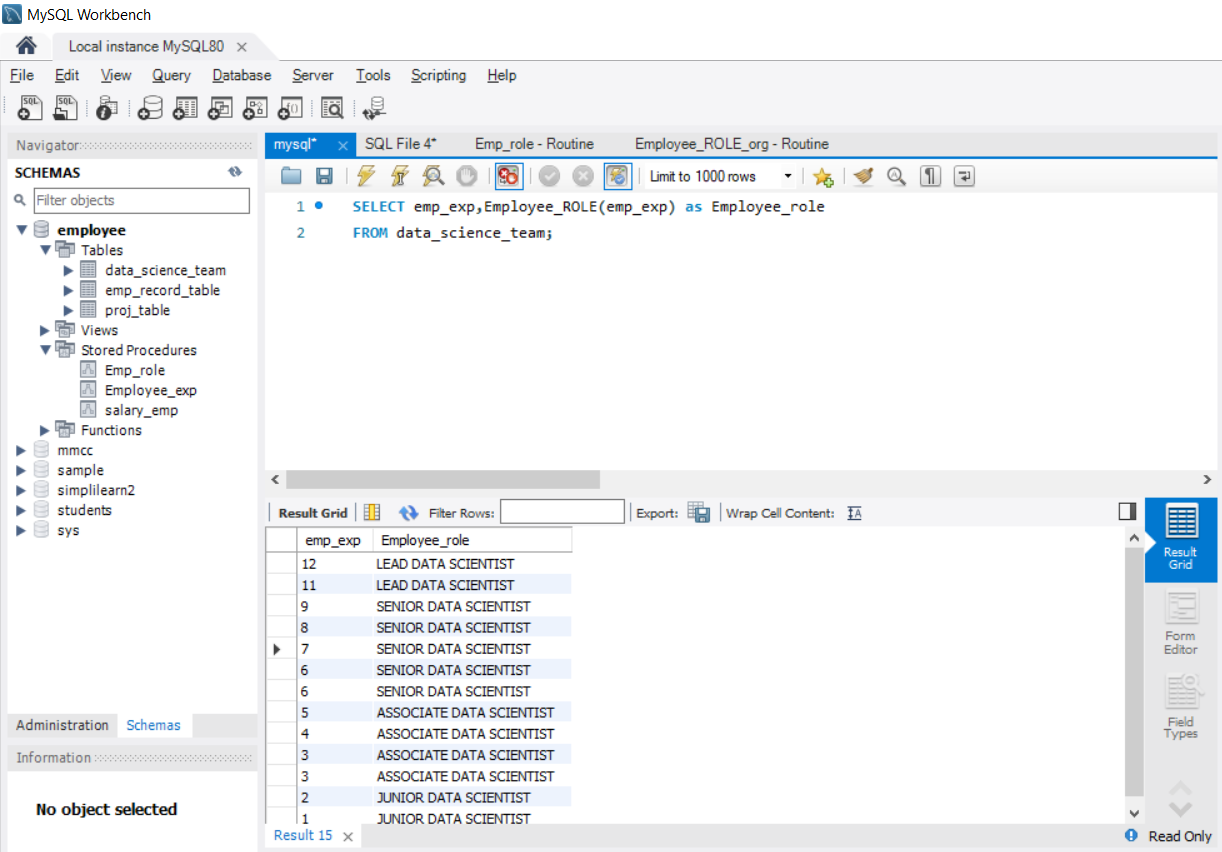
END

**Query:**

SELECT emp\_exp,Employee\_ROLE(emp\_exp) as Employee\_role

FROM data\_science\_team;

**Output:**



**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.**

**Query:**

create index first\_name\_index on emp\_record\_table(FIRST\_NAME);

show index from emp\_record\_table;

select \* from emp\_record\_table where FIRST\_NAME = 'Eric';

**Output:**

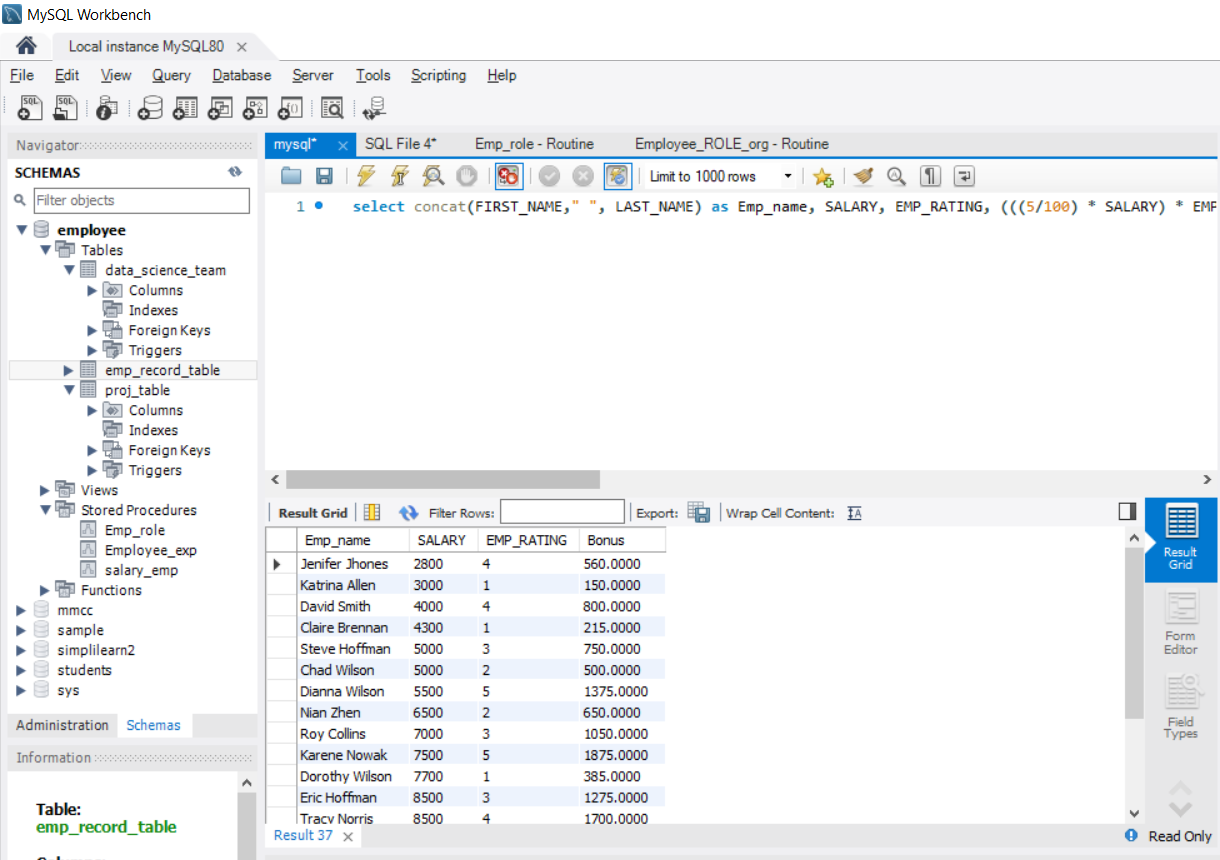


**16. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).**

**Query:**

Select concat(FIRST\_NAME," ", LAST\_NAME) as Emp\_name, SALARY, EMP\_RATING, (((5/100) \* SALARY) \* EMP\_RATING) as Bonus from emp\_record\_table;

**Output:**



**17. 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) as Average\_Salary from emp\_record\_table group by CONTINENT, COUNTRY;

**Output:**

