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.

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.

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
* greater than four
* between two and four

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.

1. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).
2. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.

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.

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.

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

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

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.

1. 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',

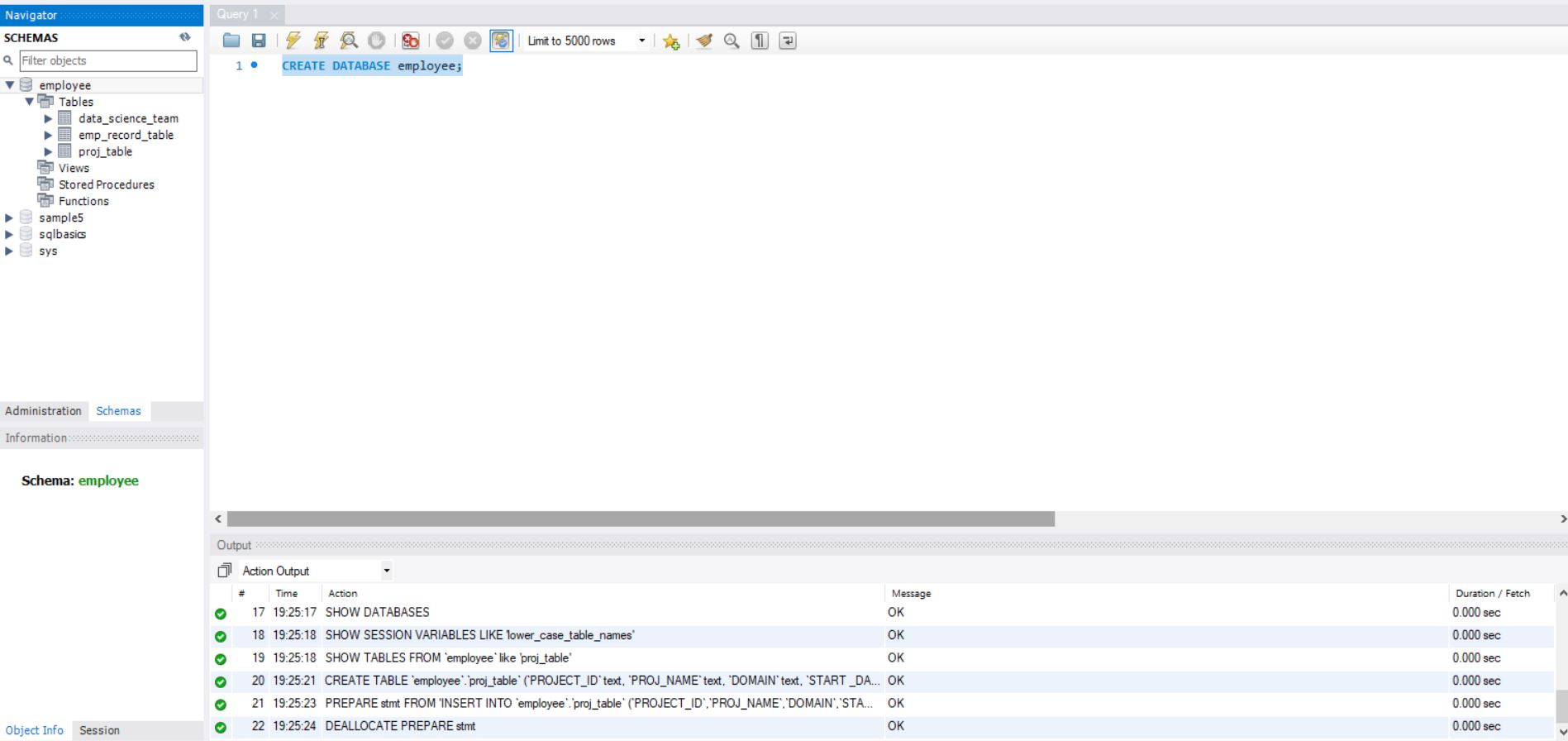
For an employee with the experience of 12 to 16 years assign 'MANAGER'.

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.

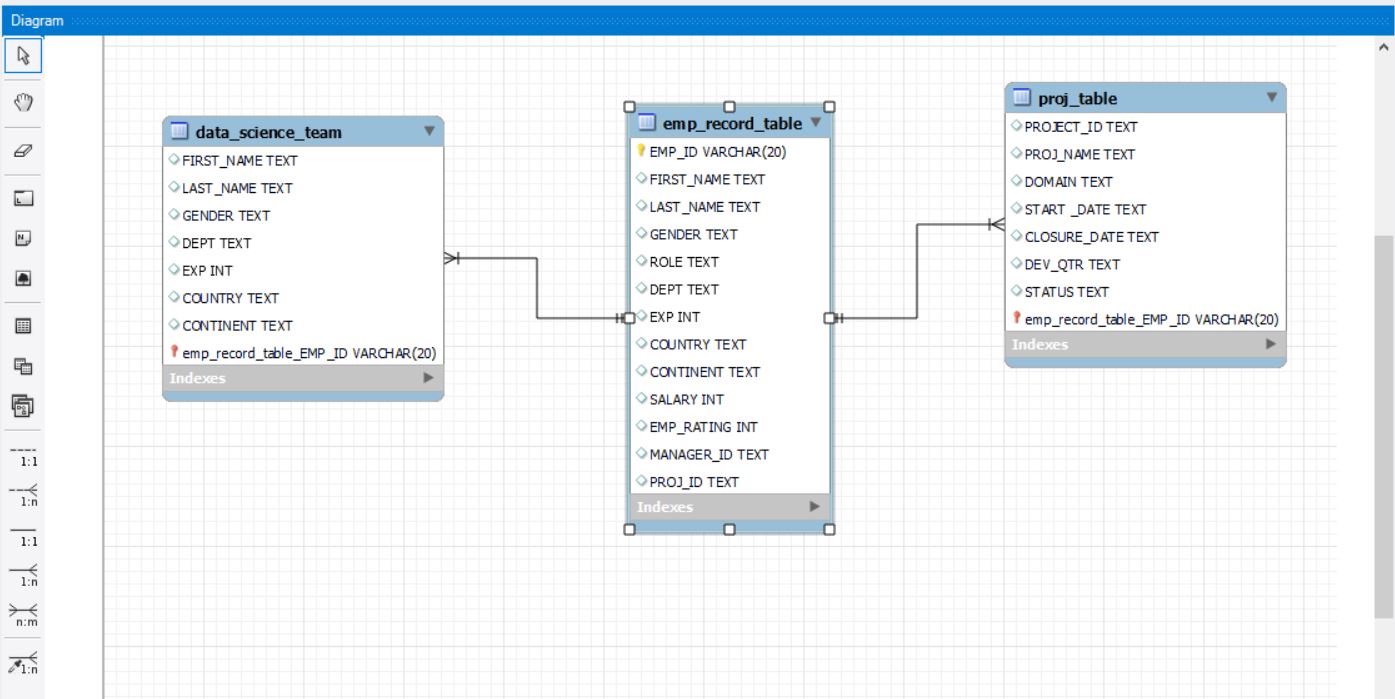
1. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).

1. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.
2. 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.

1A. CREATE DATABASE employee;



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

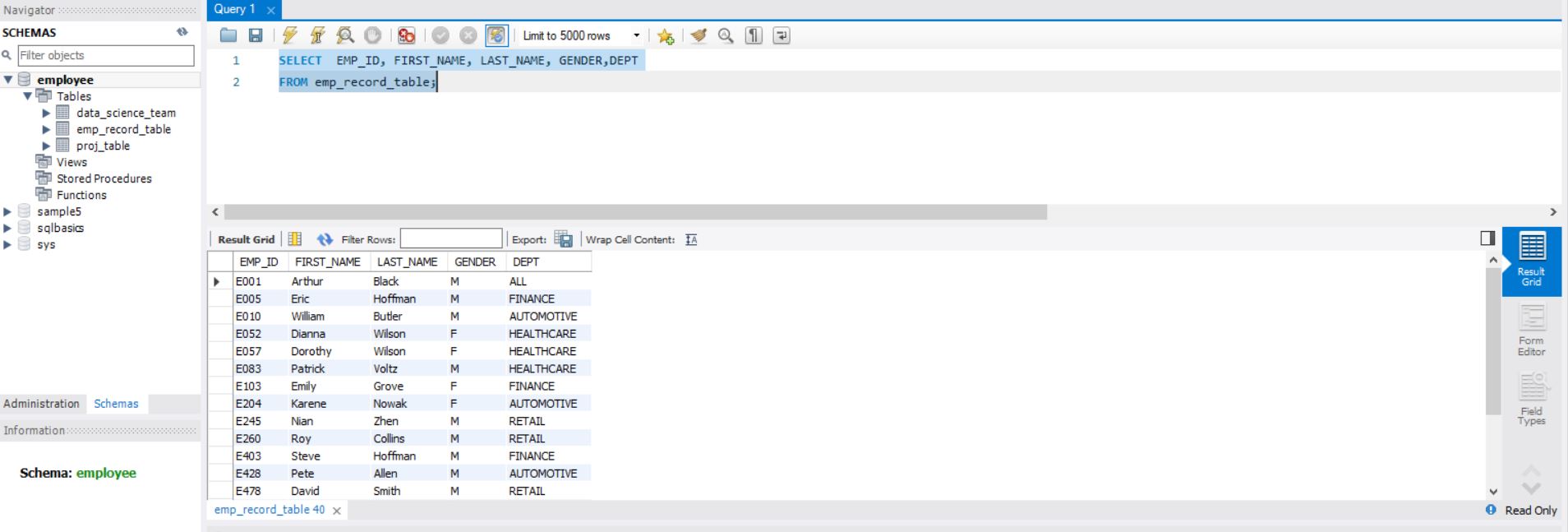


3.Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

3A.

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER,DEPT

FROM emp\_record\_table;



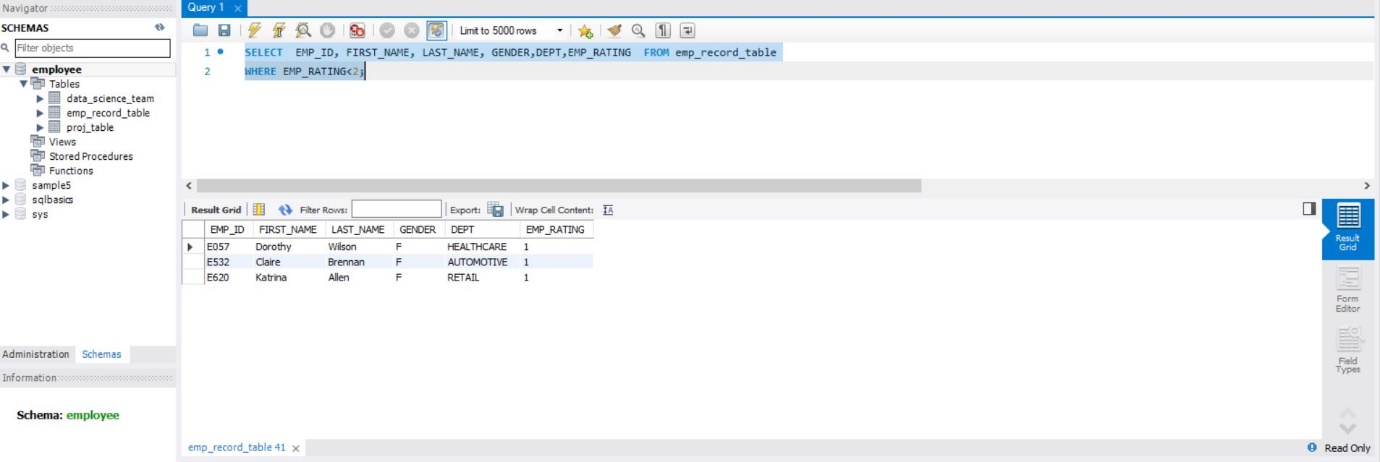
4.Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:

* less than two
* greater than four
* between two and four

4A. less than two

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER,DEPT,EMP\_RATING FROM emp\_record\_table

WHERE EMP\_RATING<2;

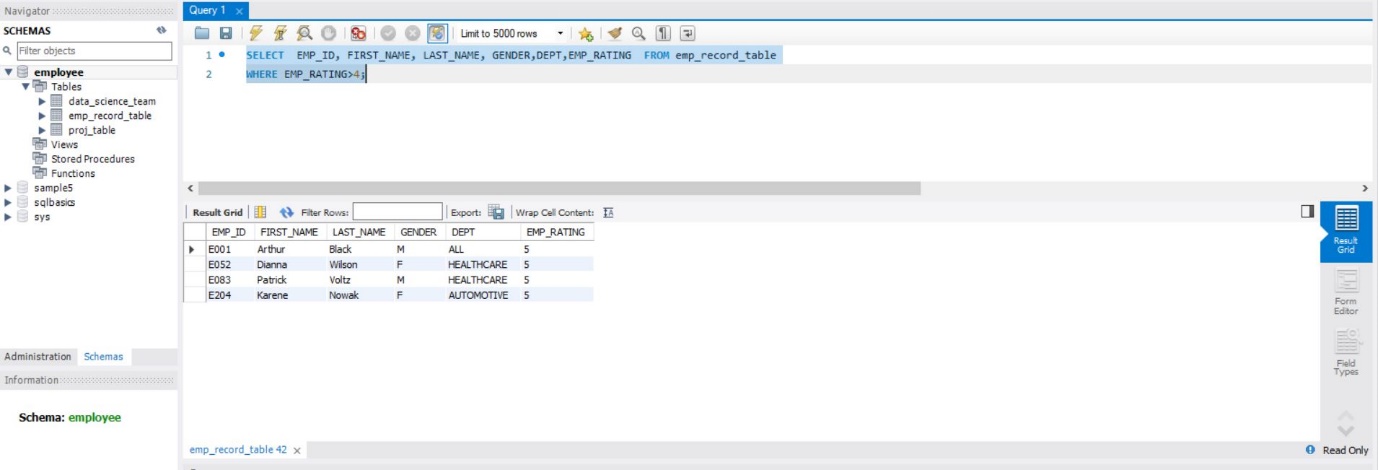


4B.

greater than four

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER,DEPT,EMP\_RATING FROM emp\_record\_table

WHERE EMP\_RATING>4;

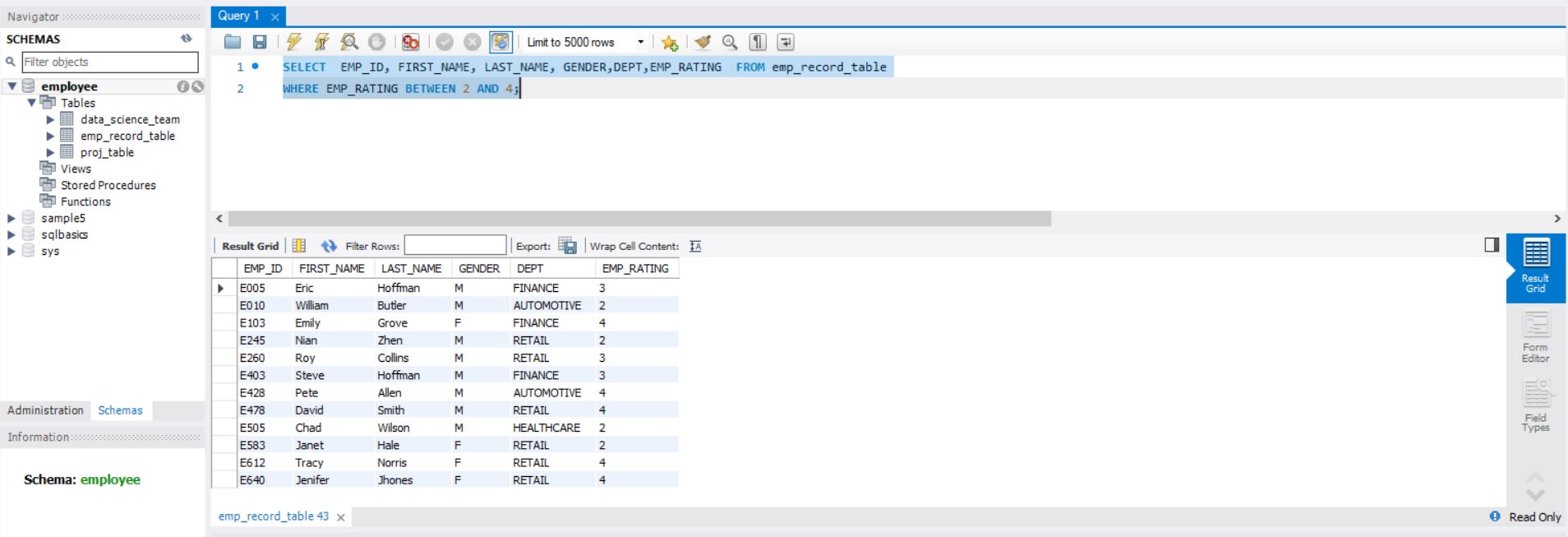


4C.

between two and four

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER,DEPT,EMP\_RATING FROM emp\_record\_table

WHERE EMP\_RATING BETWEEN 2 AND 4;



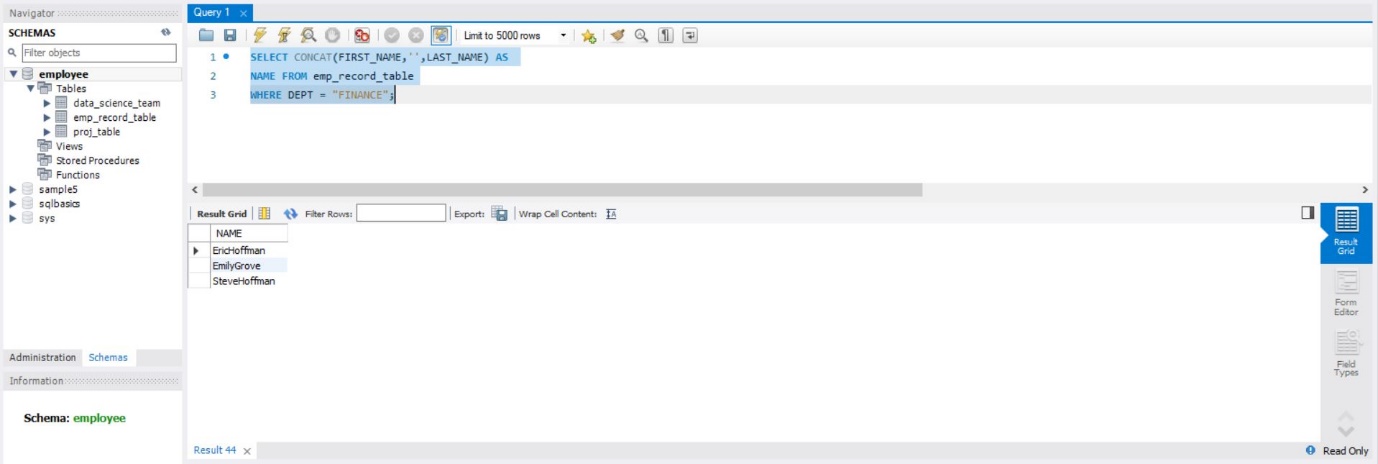
5.Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

5A.

SELECT CONCAT(FIRST\_NAME,'',LAST\_NAME) AS

NAME FROM emp\_record\_table

WHERE DEPT = "FINANCE";



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

6A.

SELECT m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.ROLE,

m.EXP,COUNT(e.EMP\_ID) as "EMP\_COUNT"

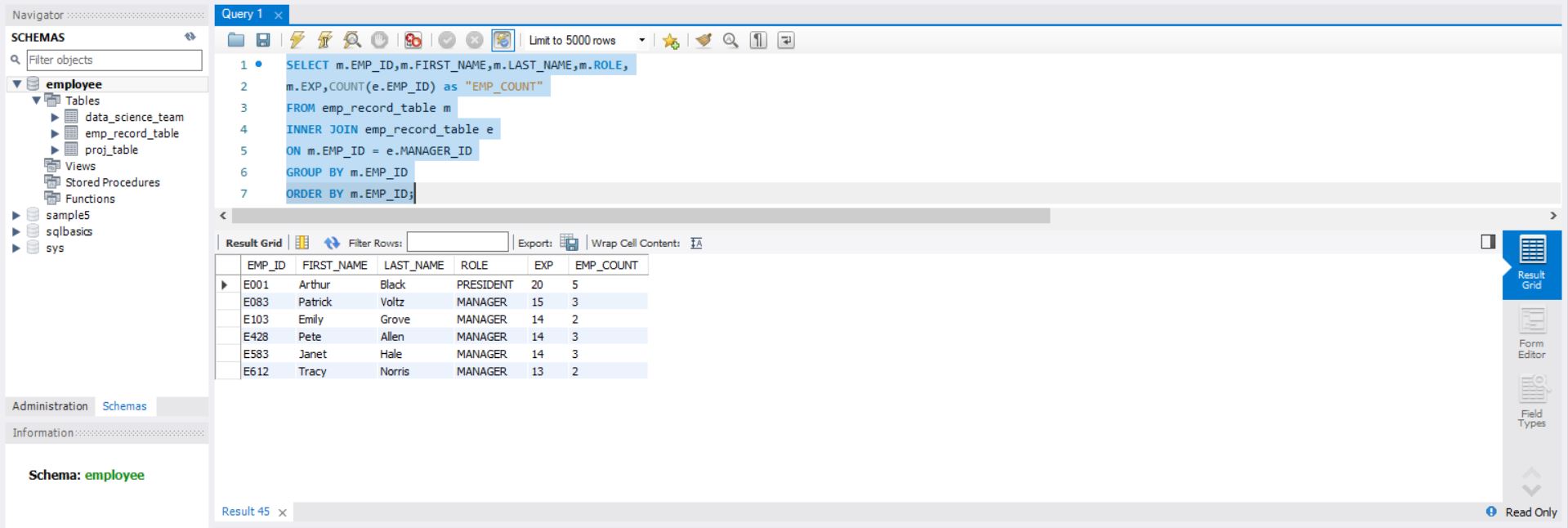
FROM emp\_record\_table m

INNER JOIN emp\_record\_table e

ON m.EMP\_ID = e.MANAGER\_ID

GROUP BY m.EMP\_ID

ORDER BY m.EMP\_ID;



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.

7A.

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,DEPT FROM emp\_record\_table

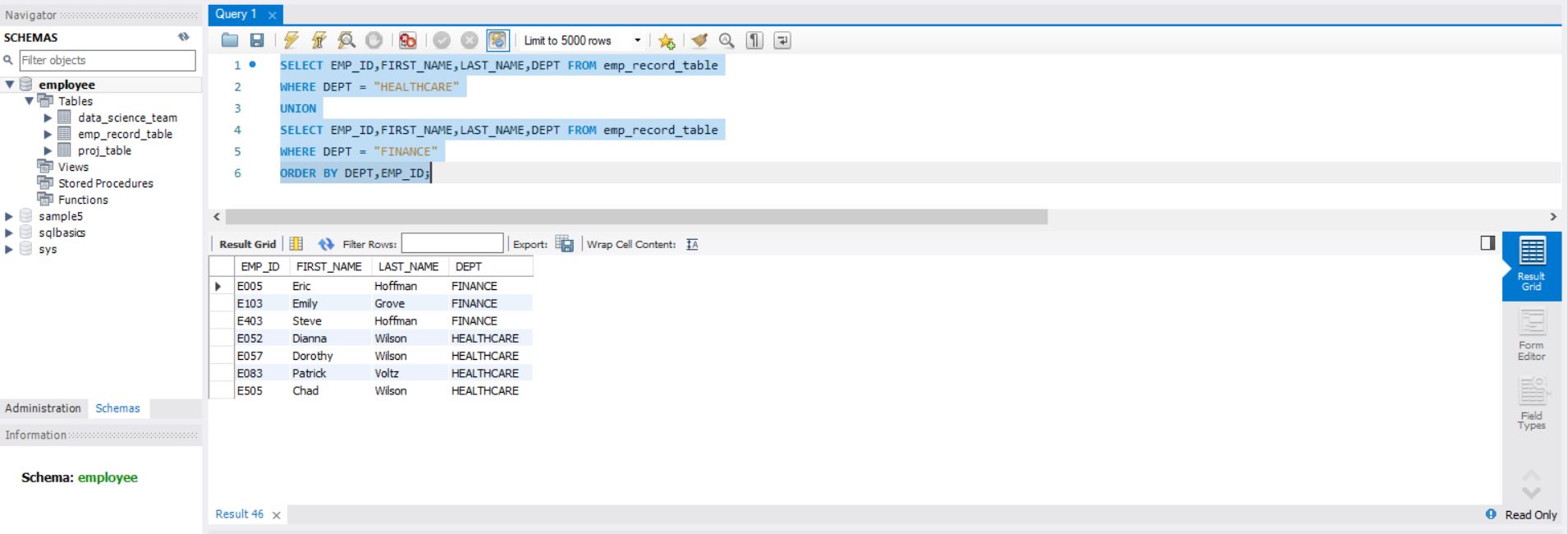
WHERE DEPT = "HEALTHCARE"

UNION

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,DEPT FROM emp\_record\_table

WHERE DEPT = "FINANCE"

ORDER BY DEPT,EMP\_ID;



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.

8A.

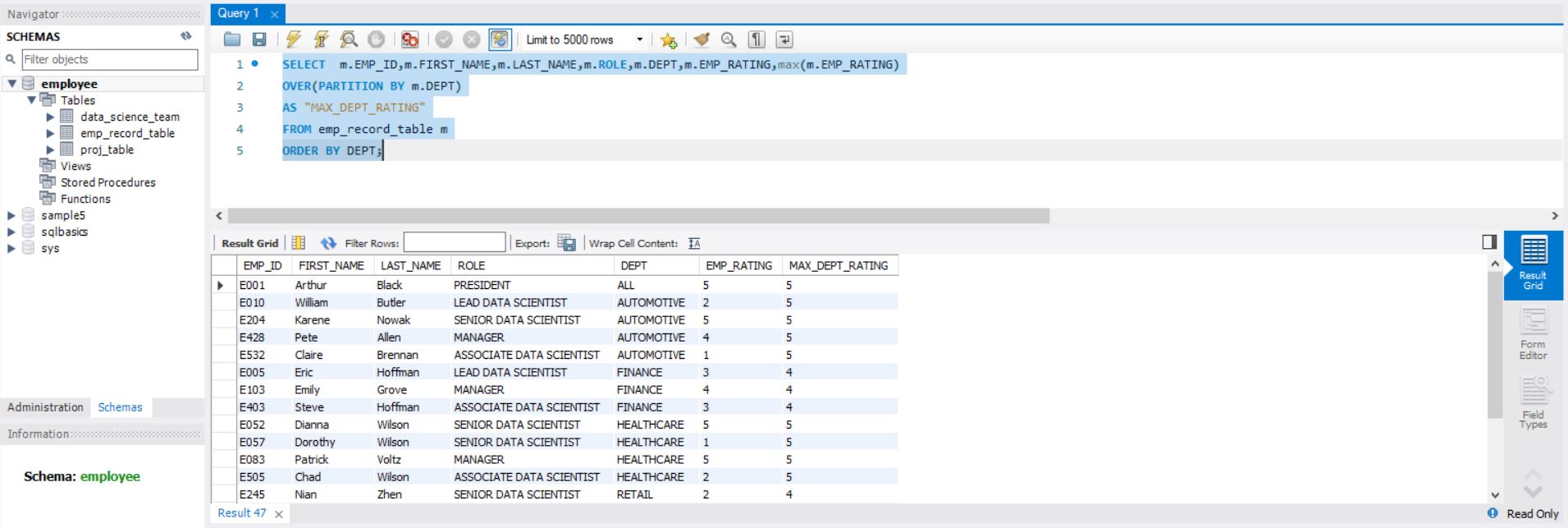
SELECT m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.ROLE,m.DEPT,m.EMP\_RATING,max(m.EMP\_RATING)

OVER(PARTITION BY m.DEPT)

AS "MAX\_DEPT\_RATING"

FROM emp\_record\_table m

ORDER BY DEPT;



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.

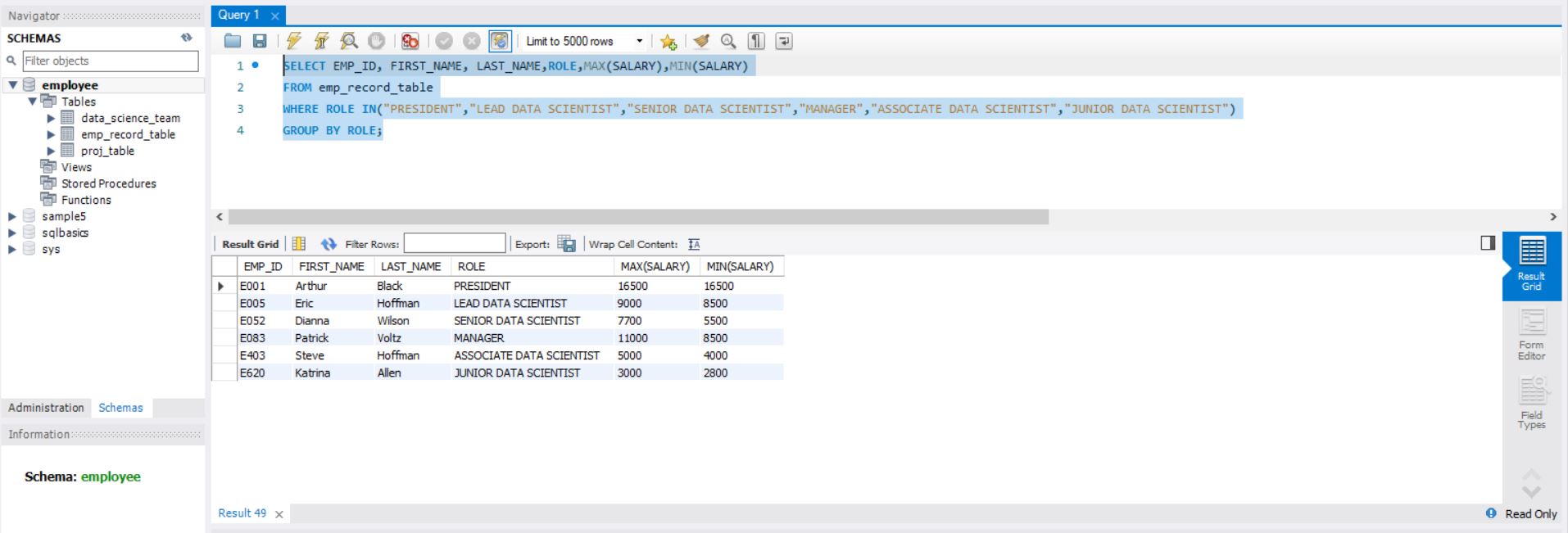
9A.

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, MAX(SALARY), MIN(SALARY)

FROM emp\_record\_table

WHERE ROLE IN("PRESIDENT","LEAD DATA SCIENTIST","SENIOR DATA SCIENTIST","MANAGER","ASSOCIATE DATA SCIENTIST","JUNIOR DATA SCIENTIST")

GROUP BY ROLE;



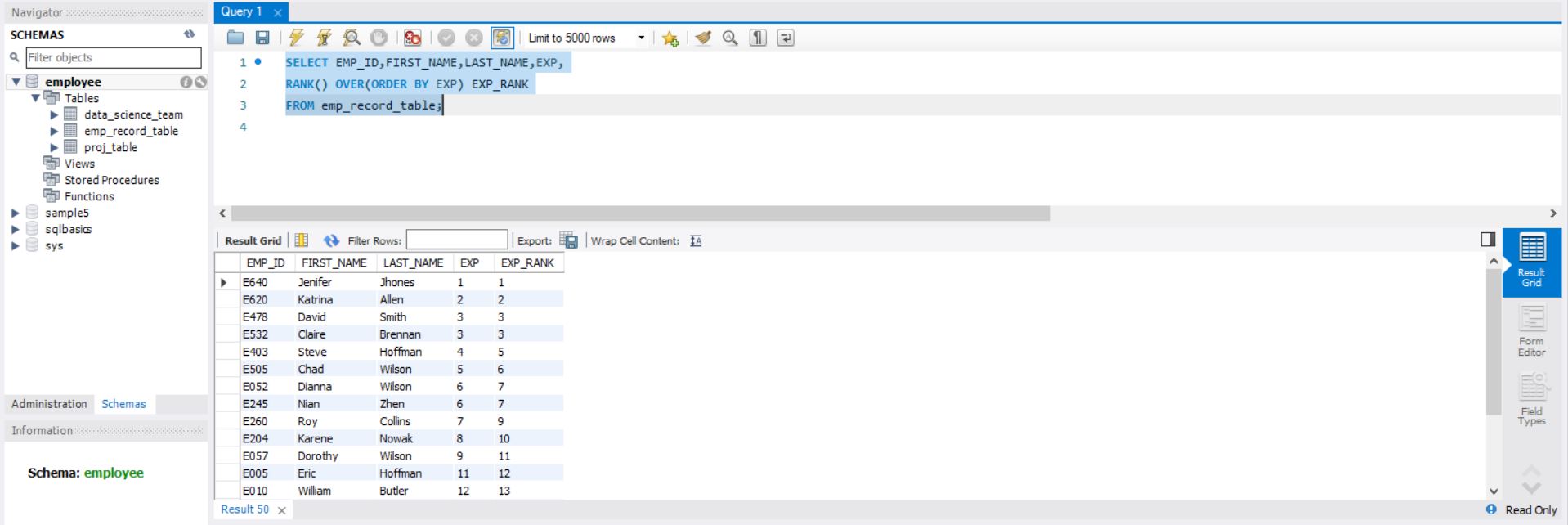
10.Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

10A.

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,EXP,

RANK() OVER(ORDER BY EXP) EXP\_RANK

FROM emp\_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.

11A.

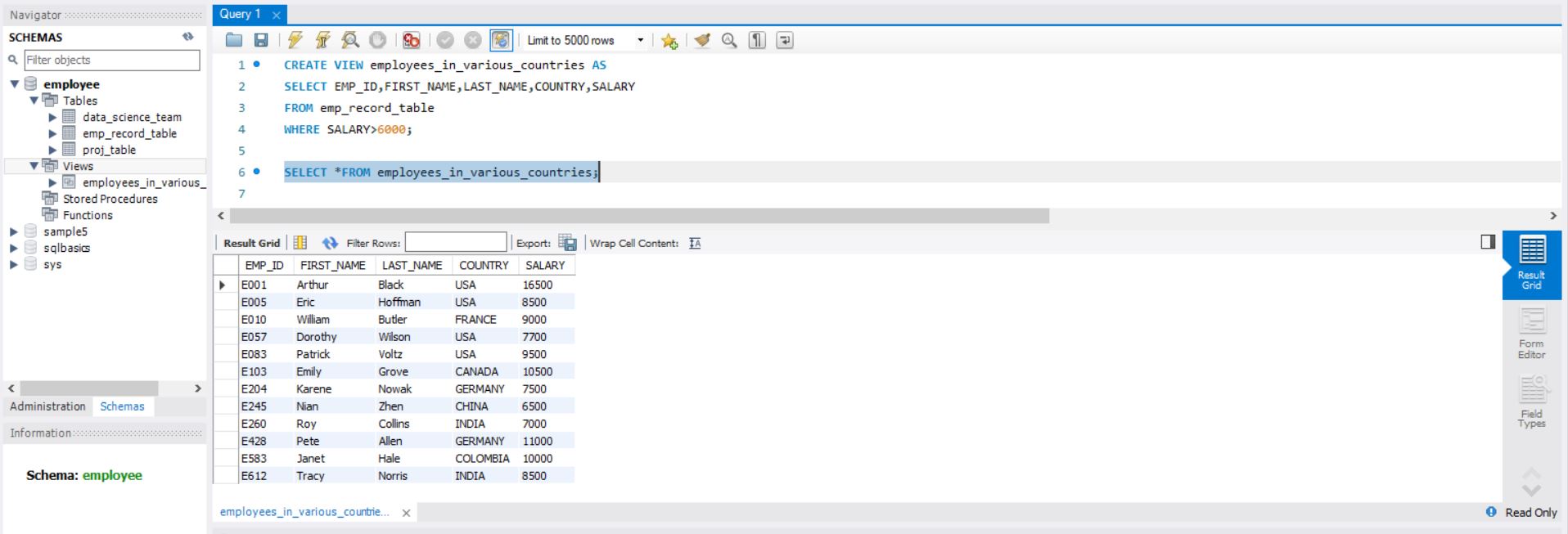
CREATE VIEW employees\_in\_various\_countries AS

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,COUNTRY,SALARY

FROM emp\_record\_table

WHERE SALARY>6000;

SELECT \*FROM employees\_in\_various\_countries;

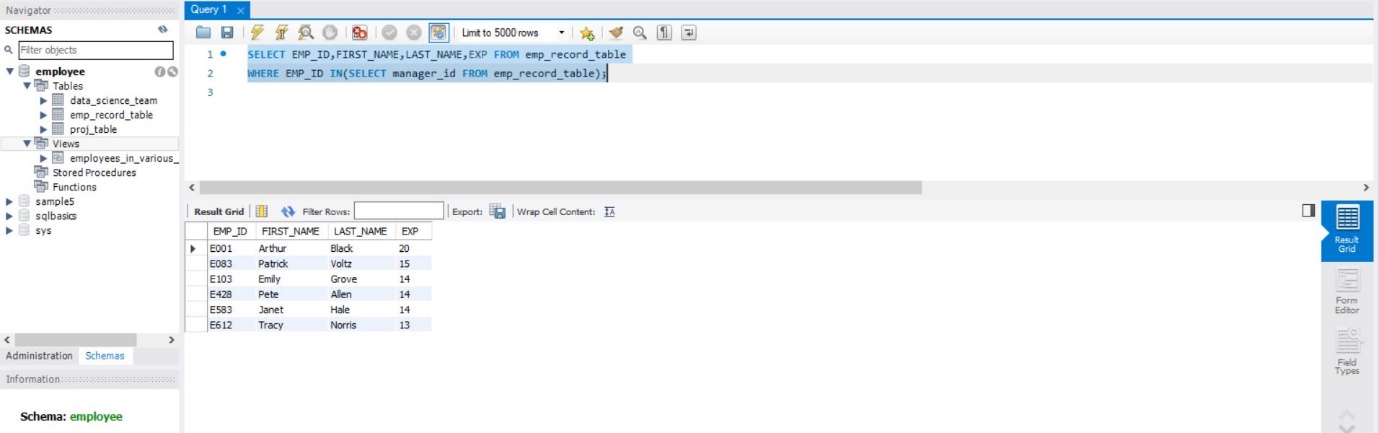


12.Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

12A.

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,EXP FROM emp\_record\_table

WHERE EMP\_ID IN(SELECT manager\_id FROM emp\_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.

13A.

DELIMITER &&

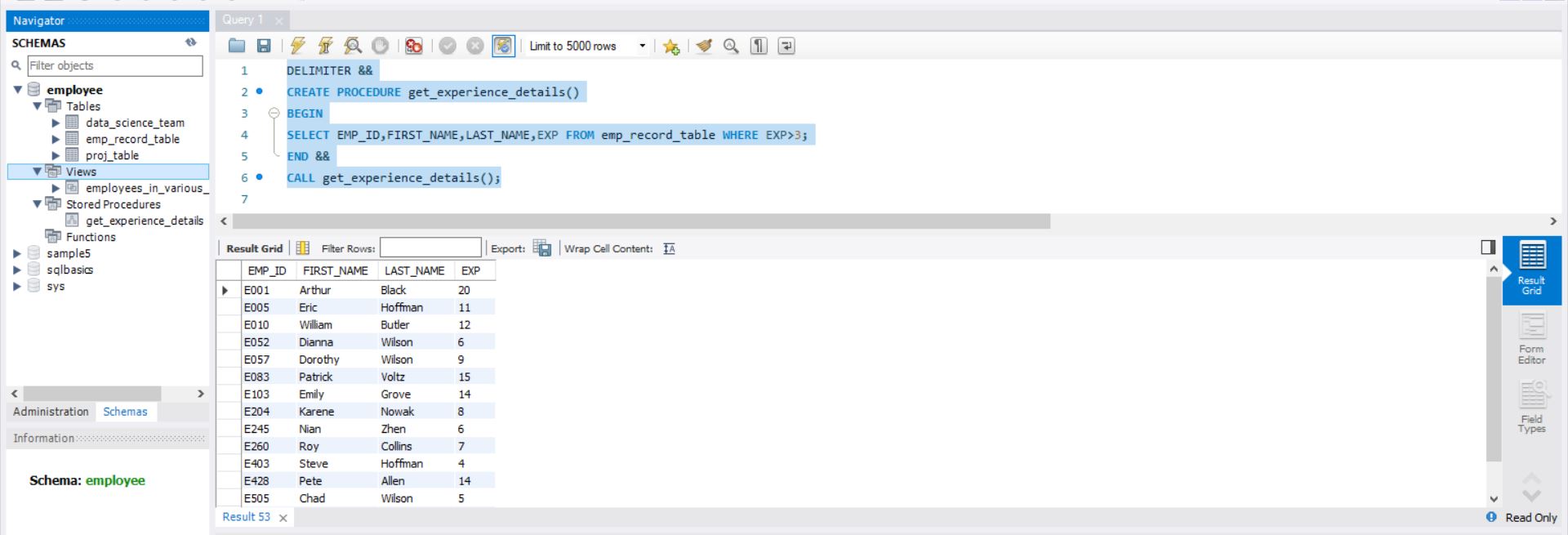
CREATE PROCEDURE get\_experience\_details()

BEGIN

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,EXP FROM emp\_record\_table WHERE EXP>3;

END &&

CALL get\_experience\_details();



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'.

14A.

DELIMITER &&

CREATE FUNCTION Employee\_ROLE(

EXP int

)

RETURNS VARCHAR(40)

DETERMINISTIC

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";

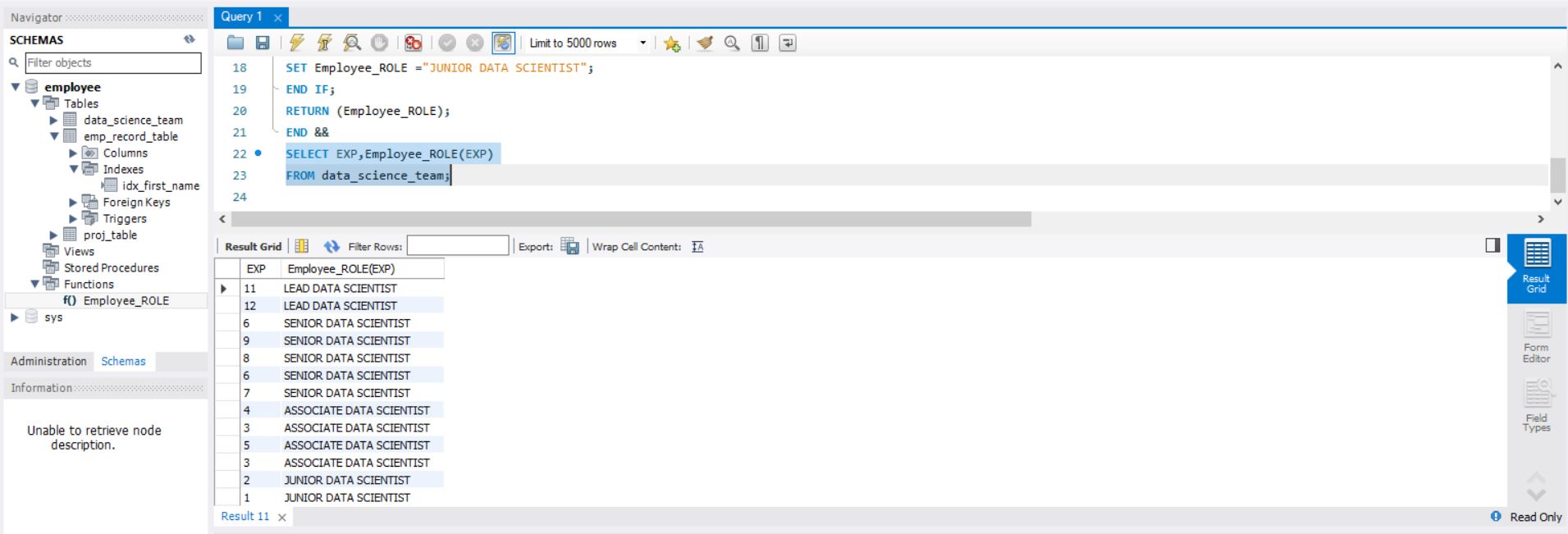
END IF;

RETURN (Employee\_ROLE);

END &&

SELECT EXP,Employee\_ROLE(EXP)

FROM data\_science\_team;



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.

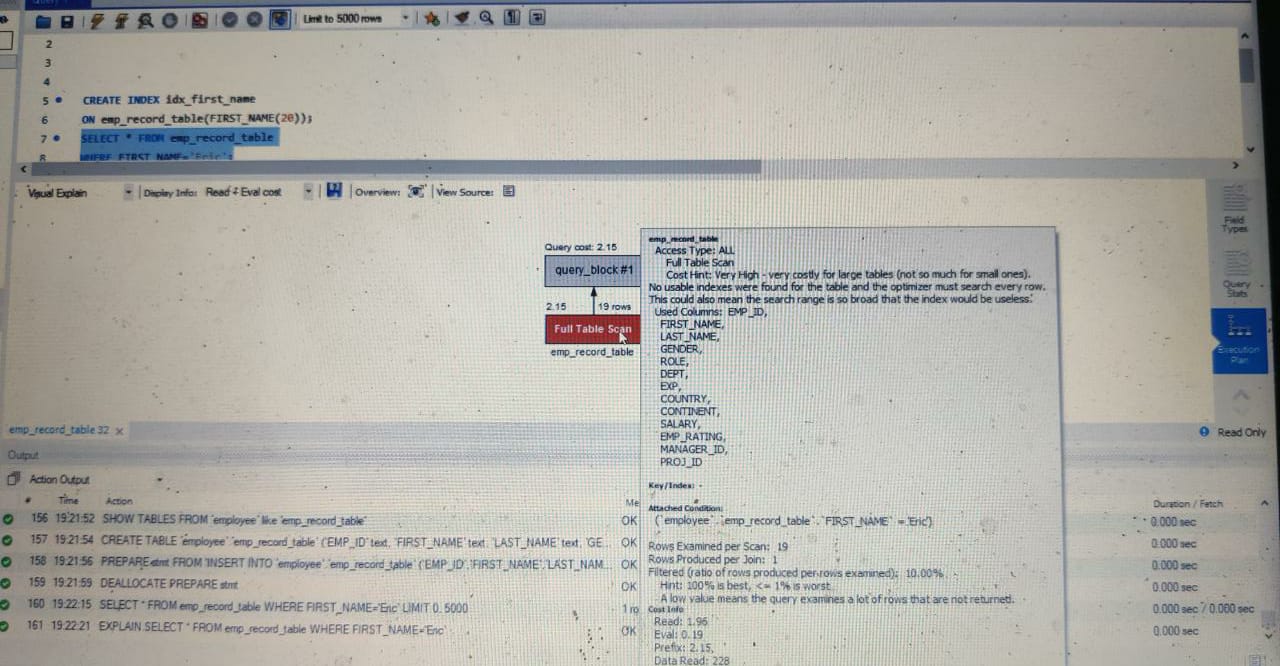
15A.

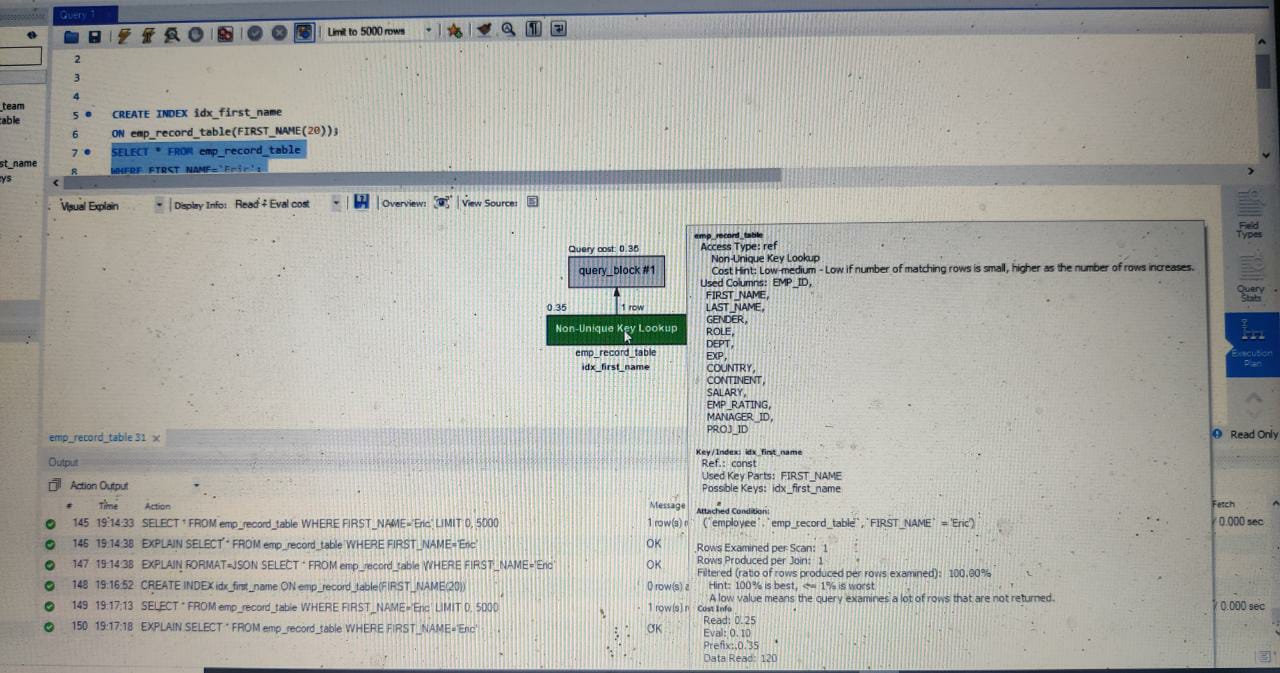
CREATE INDEX idx\_first\_name

ON emp\_record\_table(FIRST\_NAME(20));

SELECT \* FROM emp\_record\_table

WHERE FIRST\_NAME='Eric';



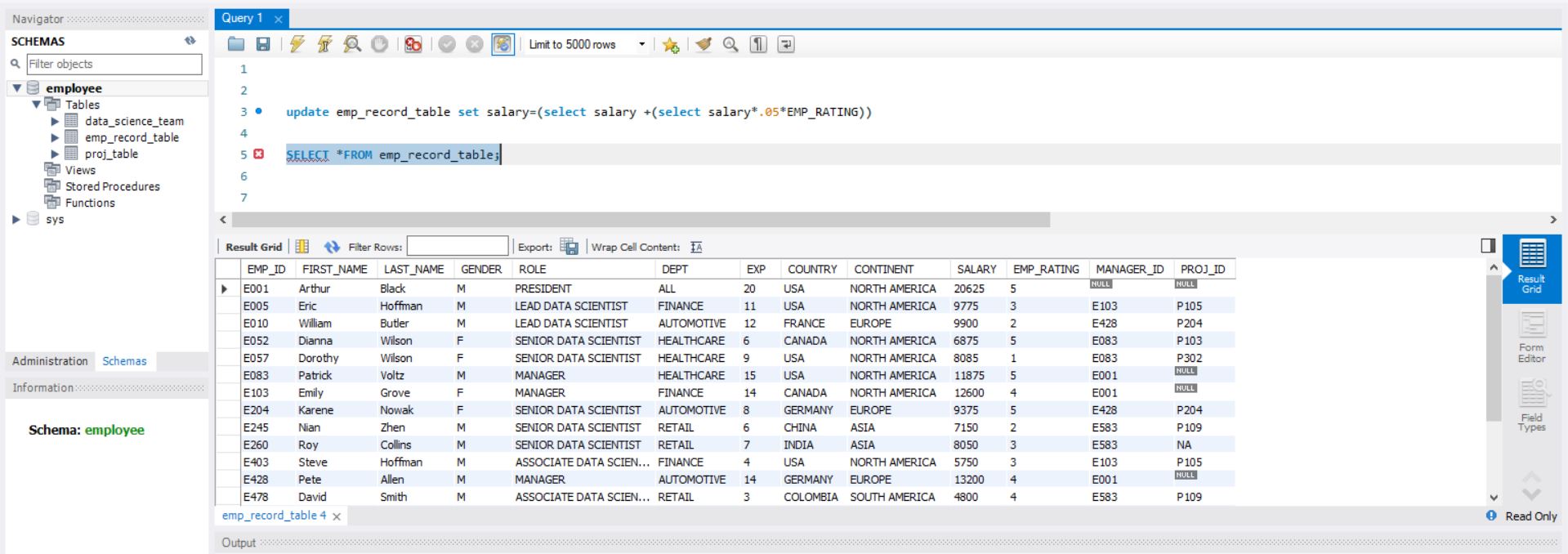


16.Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).

16A.

update emp\_record\_table set salary=(select salary +(select salary\*.05\*EMP\_RATING))

SELECT \*FROM emp\_record\_table;



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

17A.

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,SALARY,COUNTRY,CONTINENT,

AVG(salary)OVER(PARTITION BY COUNTRY)AVG\_salary\_IN\_COUNTRY,

AVG(salary)OVER(PARTITION BY CONTINENT)AVG\_salary\_IN\_CONTINENT,

COUNT(\*)OVER(PARTITION BY COUNTRY)COUNT\_IN\_COUNTRY,

COUNT(\*)OVER(PARTITION BY CONTINENT)COUNT\_IN\_CONTINENT

FROM emp\_record\_table;

