ScienceQtech Employee Performance Mapping

RAJEEV VHANHUVE

Date

6/19/2022

Course title

SQL

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.

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 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'.

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.

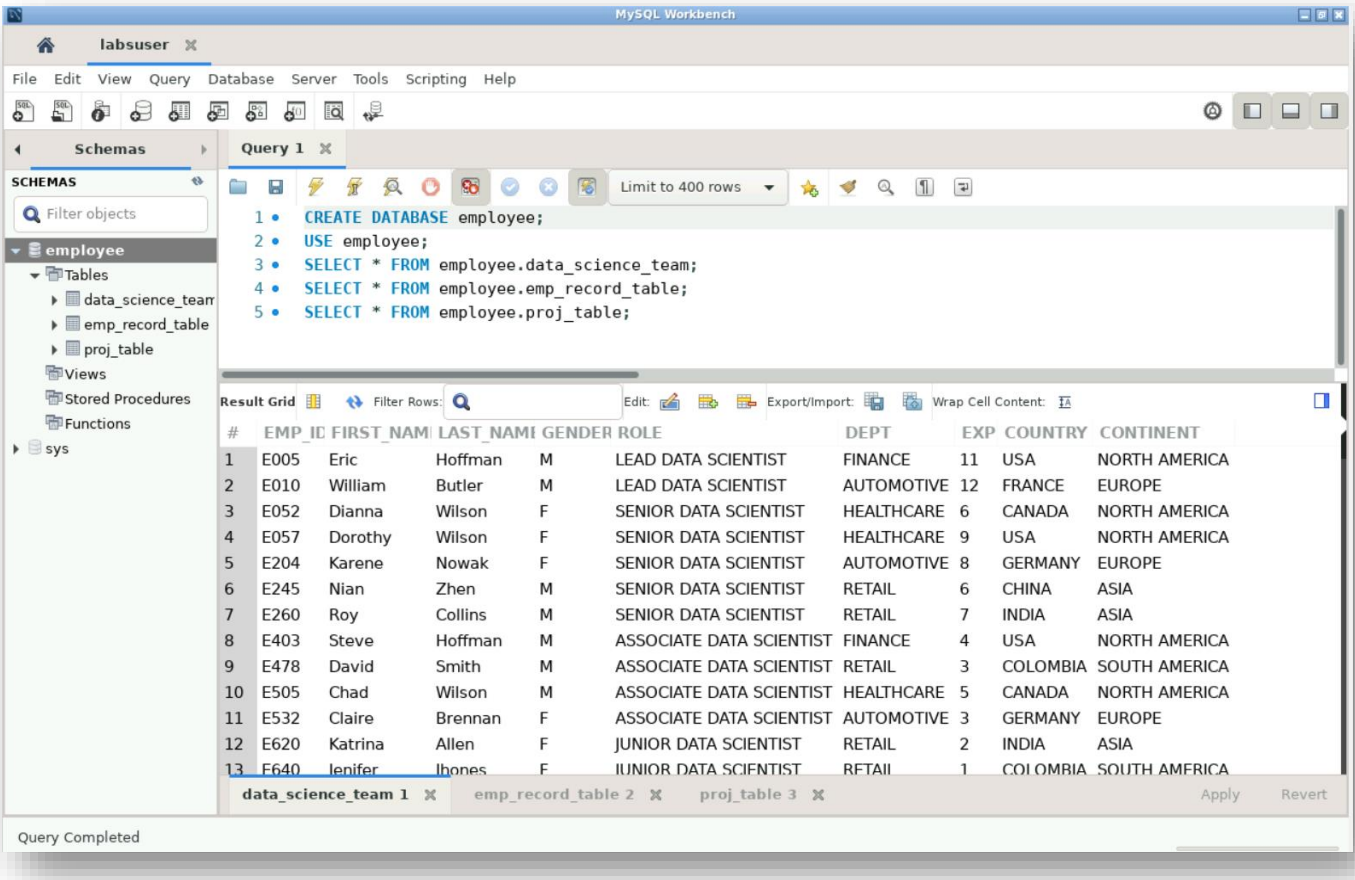
SOLUTION-

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.

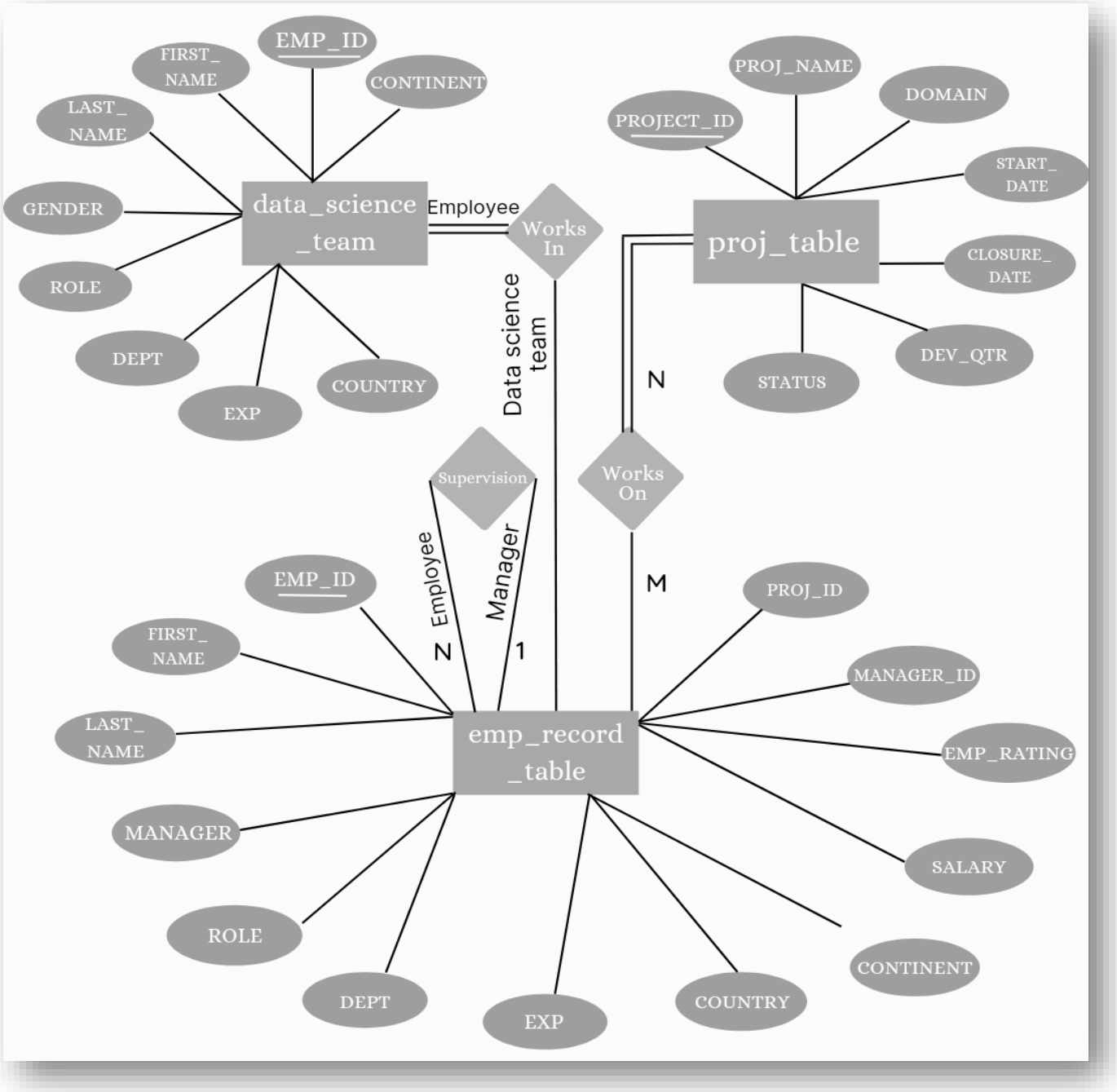
Query-

```
CREATE DATABASE employee;
USE employee;

#imported emp_record_table.csv , data_science_team.csv and proj_table.csv using Table Data Import Wizard option.

SELECT *FROM emp_record_table;
SELECT *FROM data_science_team;
SELECT *FROM proj_table;
```

Output-



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

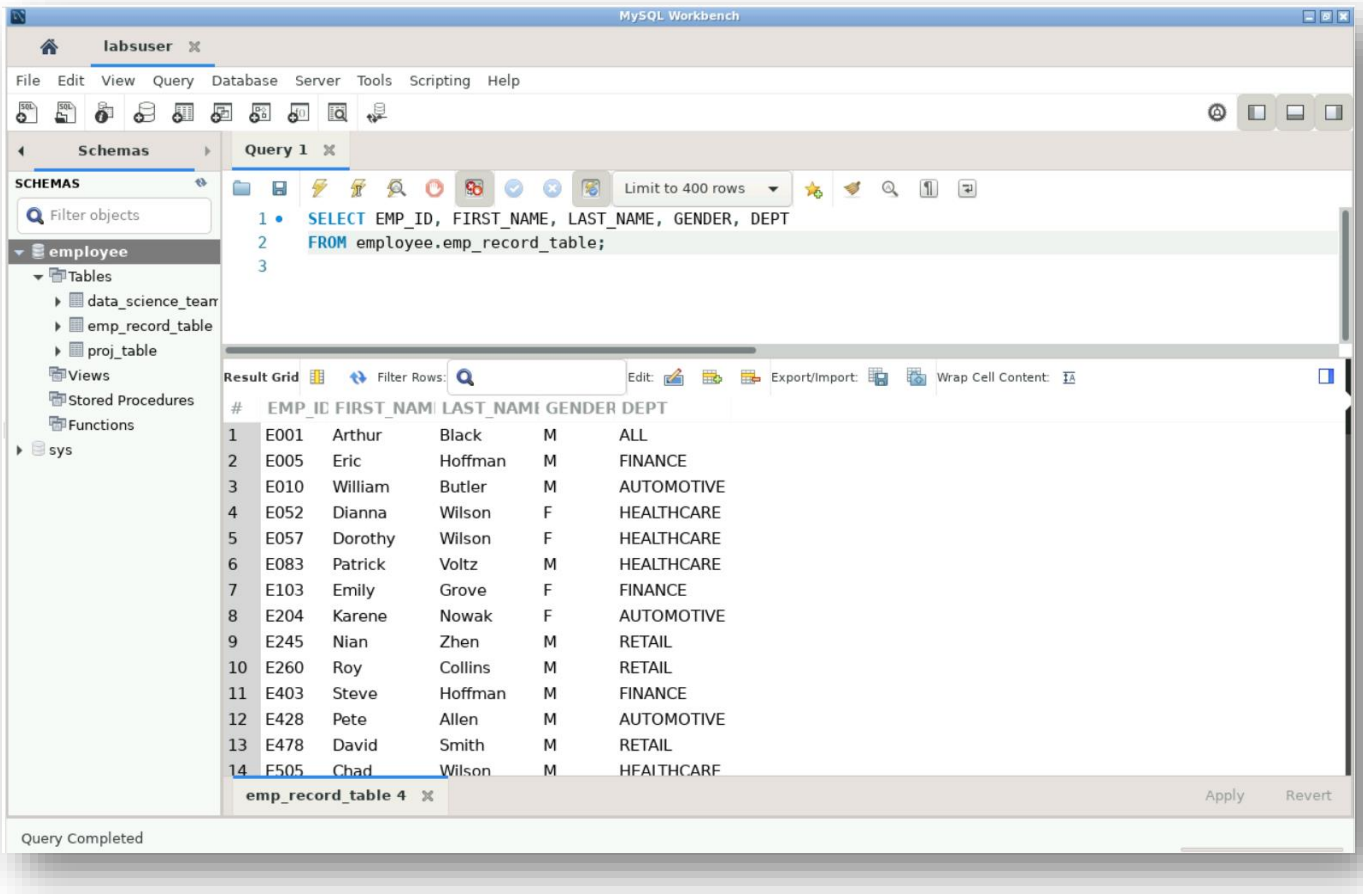


3. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

Query-

```
SELECT  EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM employee.emp_record_table;
```

Output-



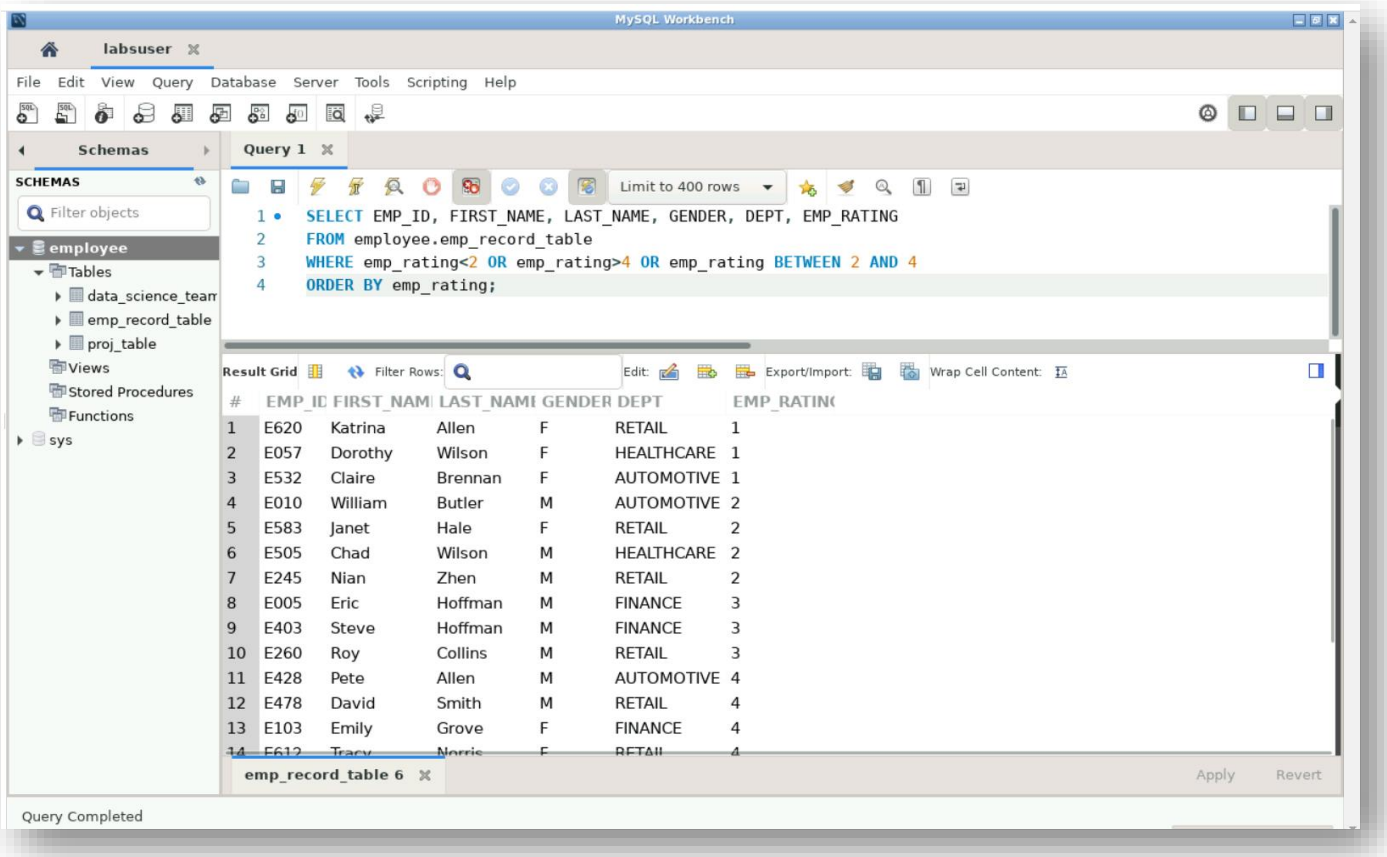
4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

- less than two
- greater than four
- between two and four

Query-

```
SELECT  EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM employee.emp_record_table
WHERE emp_rating < 2  OR  emp_rating > 4  OR  emp_rating BETWEEN  2  AND  4
ORDER BY emp_rating;
```

Output-

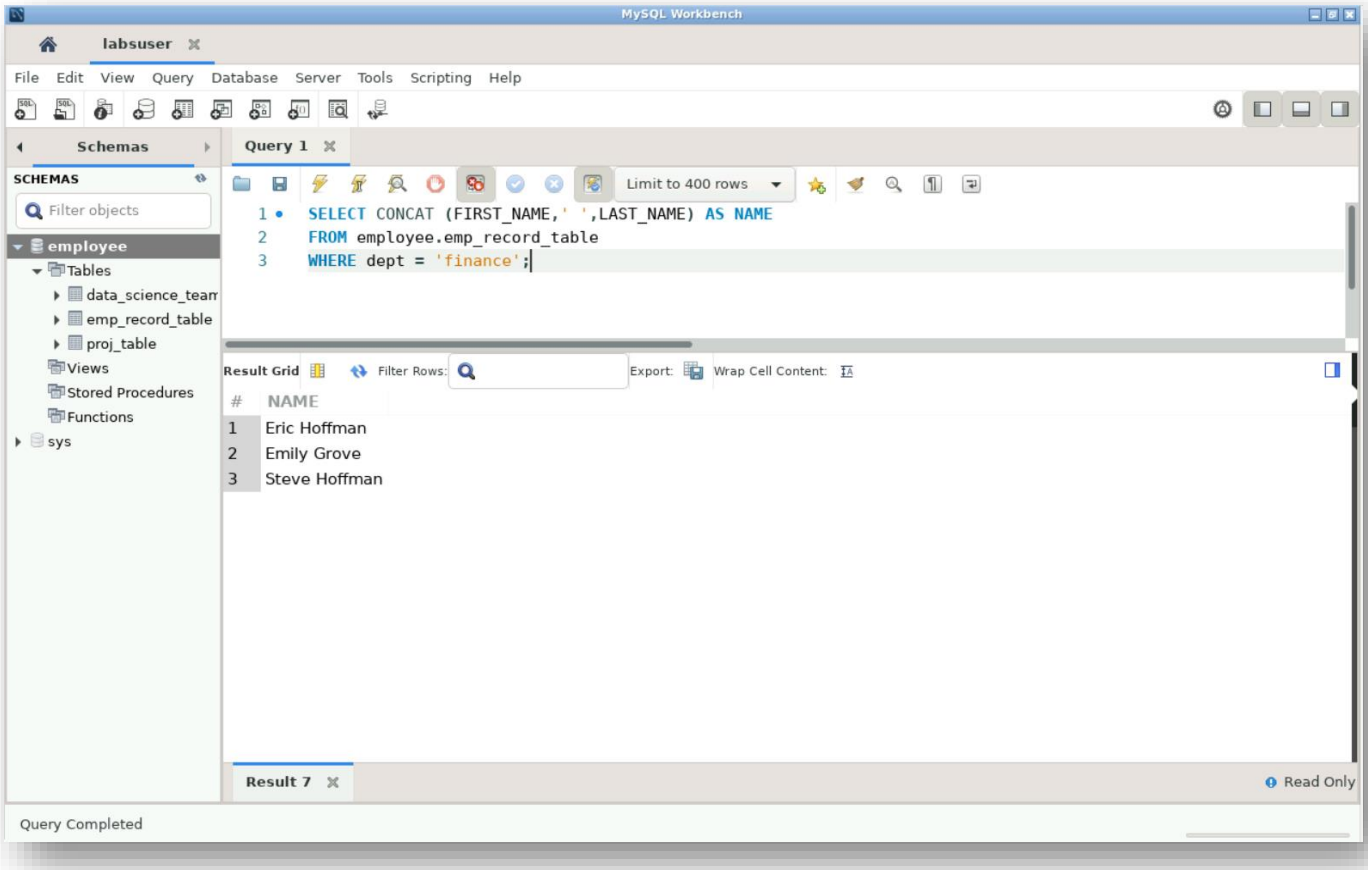


5. Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

Query-

```
SELECT  CONCAT(FIRST_NAME,’ ‘, LAST_NAME) AS NAME
FROM employee.emp_record_table
WHERE dept = ‘finance’;
```

Output-

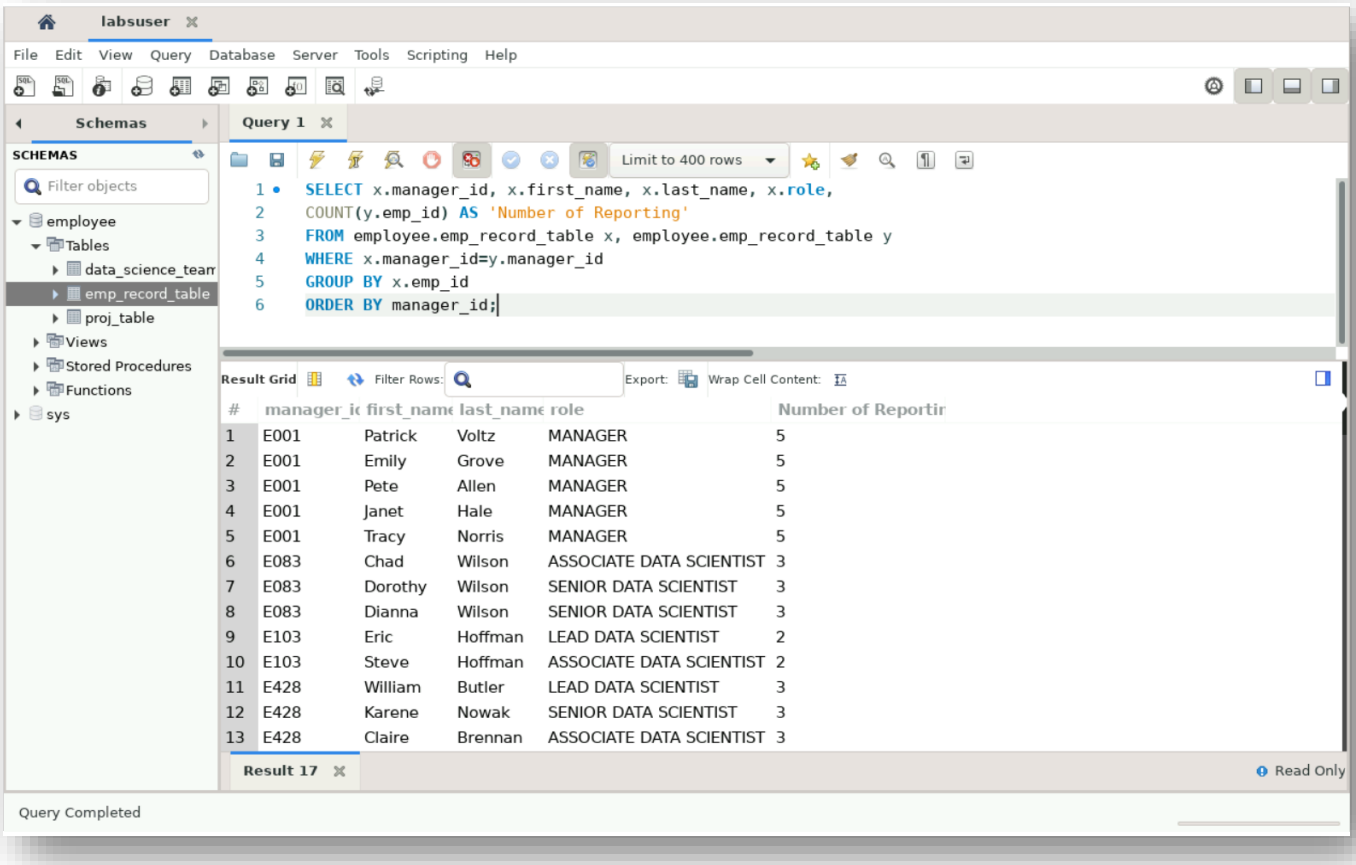


6. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).

Query-

```
SELECT x.manager_id, x.first_name, x.last_name, x.role,
COUNT(y.emp_id) AS 'Number of Reporting'
FROM employee.emp_record_table x, employee.emp_record_table y
WHERE x.manager_id=y.manager_id
GROUP BY x.emp_id
ORDER BY manager_id;
```

Output-

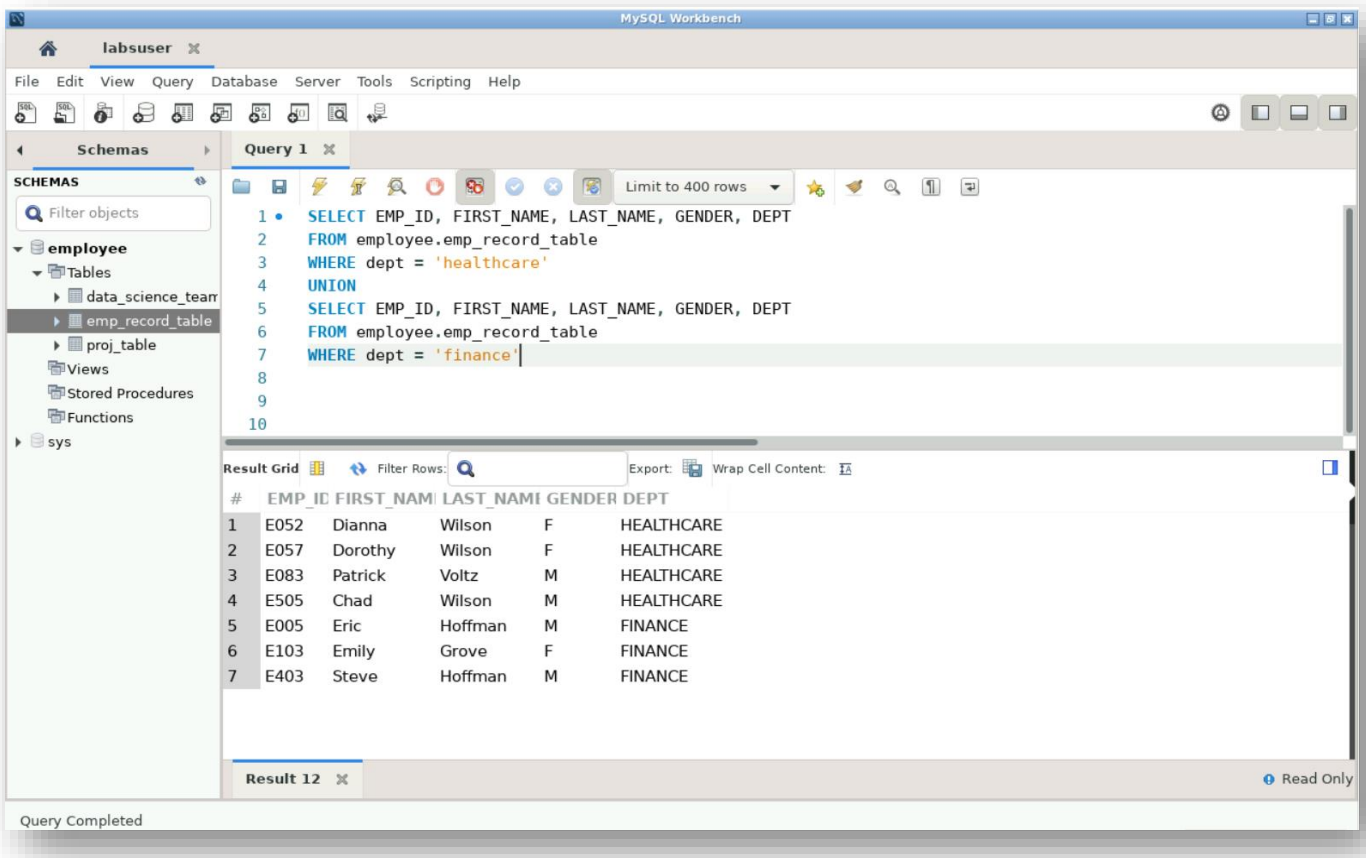


7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.

Query-

```
SELECT emp_id, first_name, last_name, gender, dept
FROM employee.emp_record_table
WHERE dept='healthcare'
UNION
SELECT emp_id, first_name, last_name, gender, dept
FROM employee.emp_record_table
WHERE dept=' finance';
```

Output-

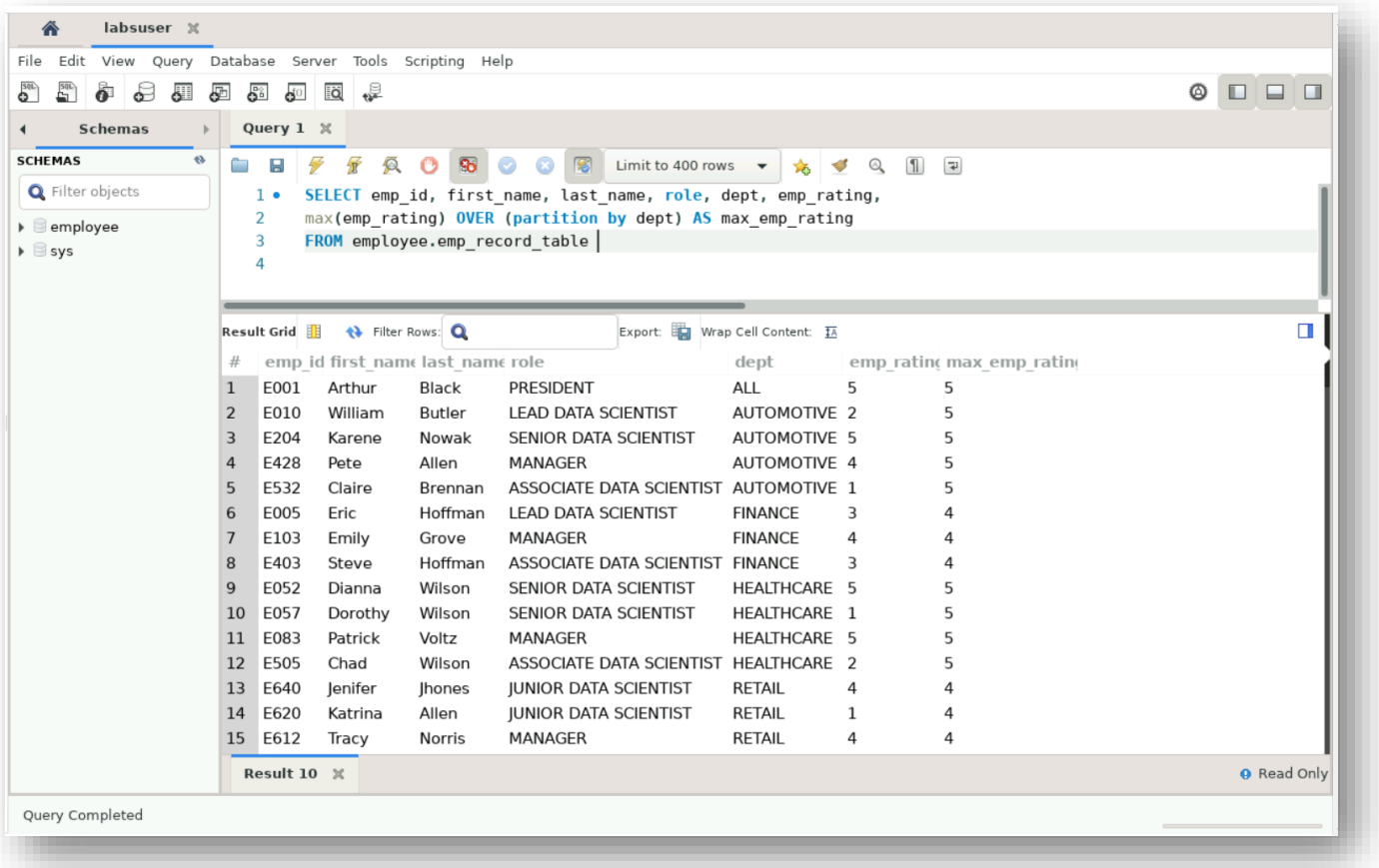


8. Write a query to list down employee details such as EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING grouped by dept. Also, include the respective employee rating along with the max emp rating for the department.

Query-

```
SELECT emp_id, first_name, last_name, emp_rating, dept, emp_rating,
MAX(emp_rating) OVER (partition by dept) AS max_emp_rating
FROM employee.emp_record_table
```

Output-

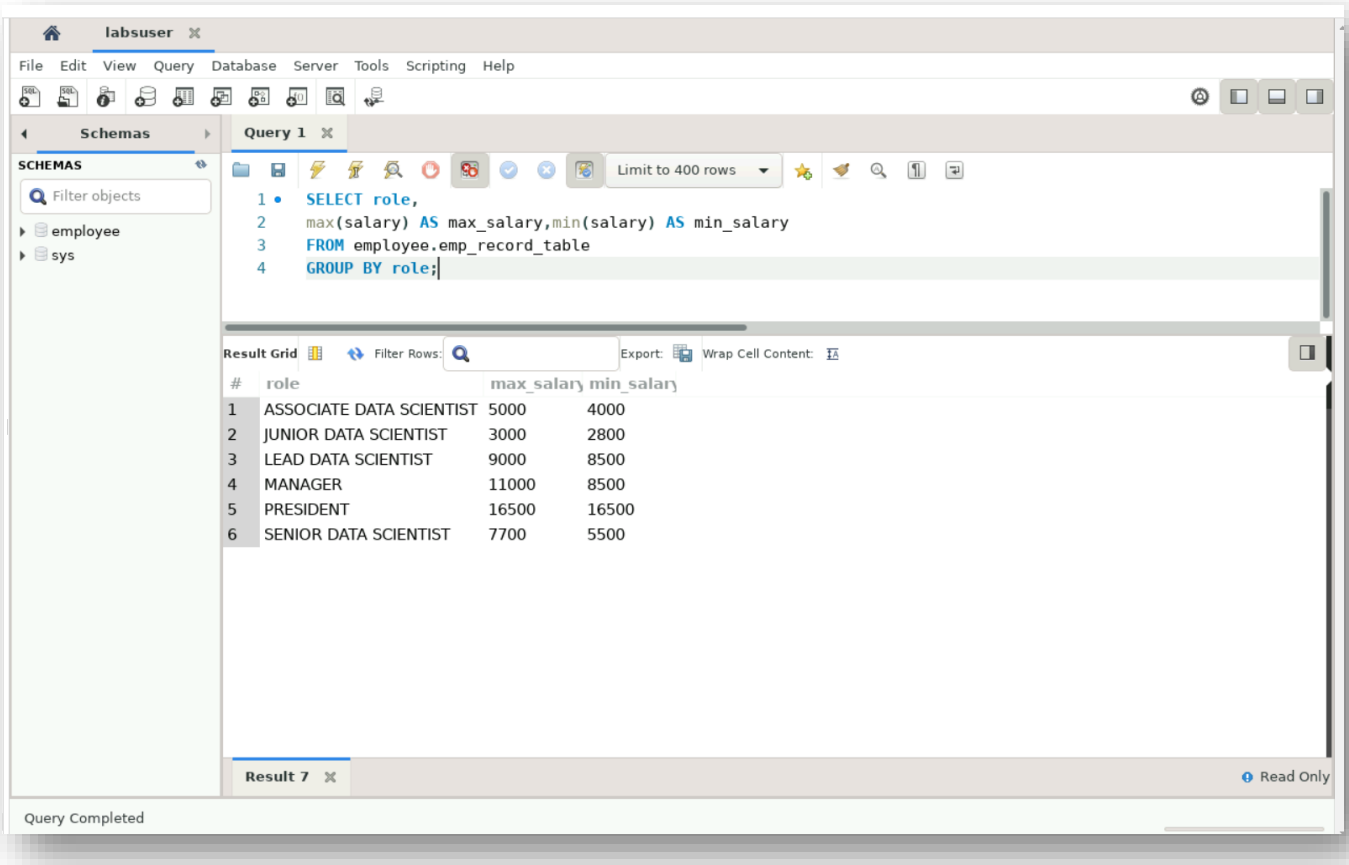


9. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

Query-

```
SELECT role,
MAX(salary) AS max_salary, MIN(salary) AS min_salary
FROM employee.emp_record_table
GROUP BY role;
```

Output-

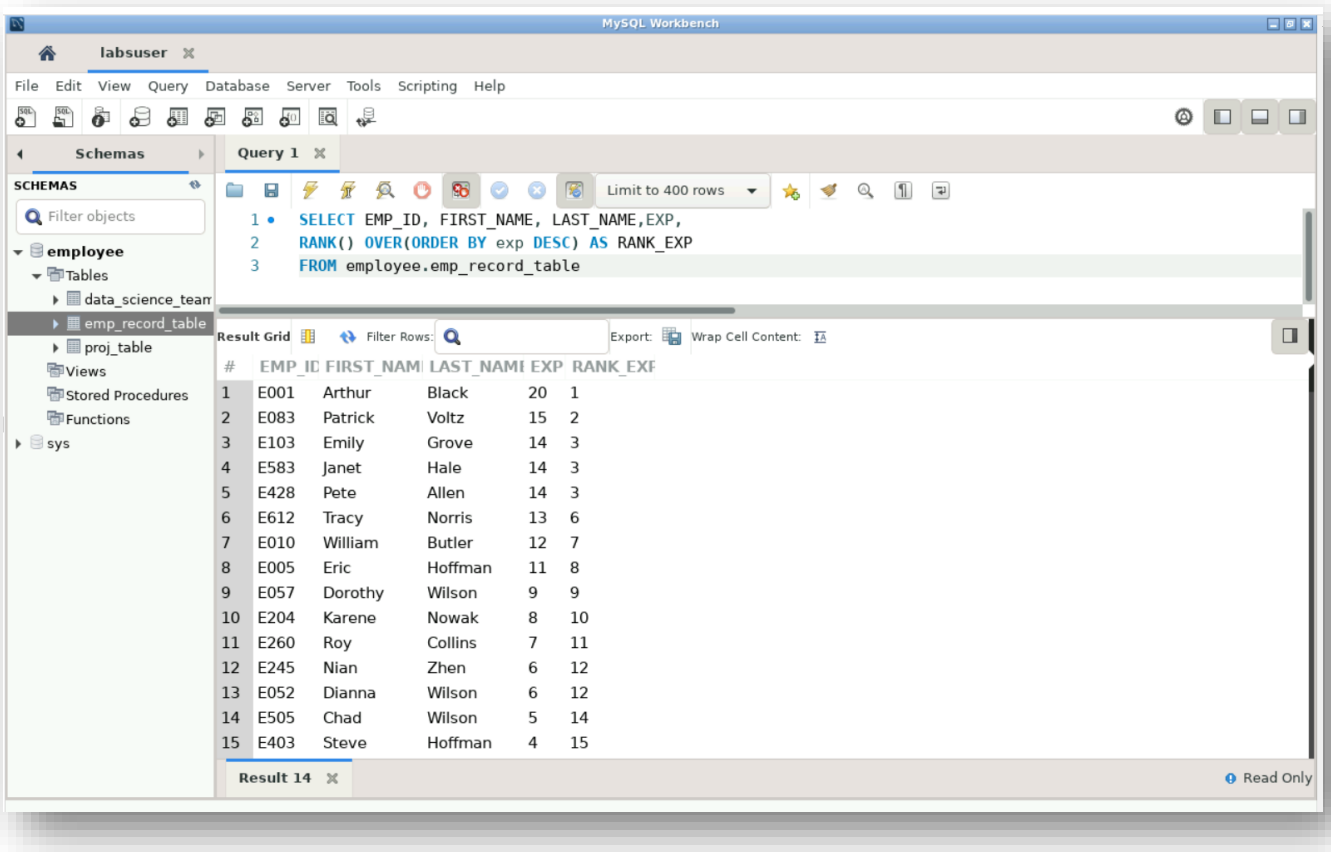


10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

Query-

```
SELECT emp_id, first_name, last_name, exp,
RANK( ) OVER(ORDER BY exp DESC) AS RANK_EXP
FROM employee.emp_record_table
```

Output-

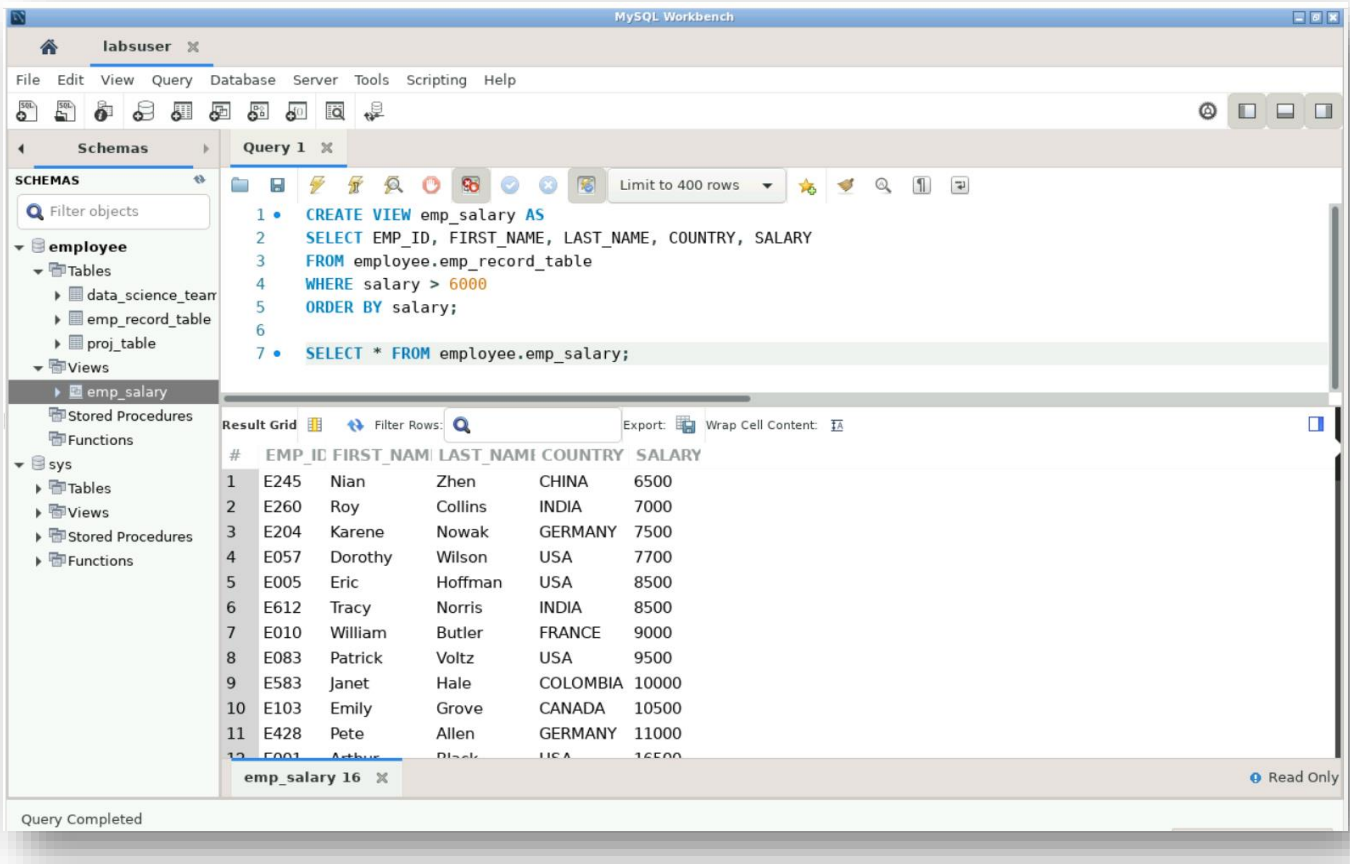


11. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.

Query-

```
CREATE VIEW emp_salary AS
SELECT emp_id, first_name, last_name, country, salary
FROM employee.emp_record_table
WHERE salary > 6000
GROUP BY country
ORDER BY salary;

SELECT* FROM emp_salary;
```

Output-

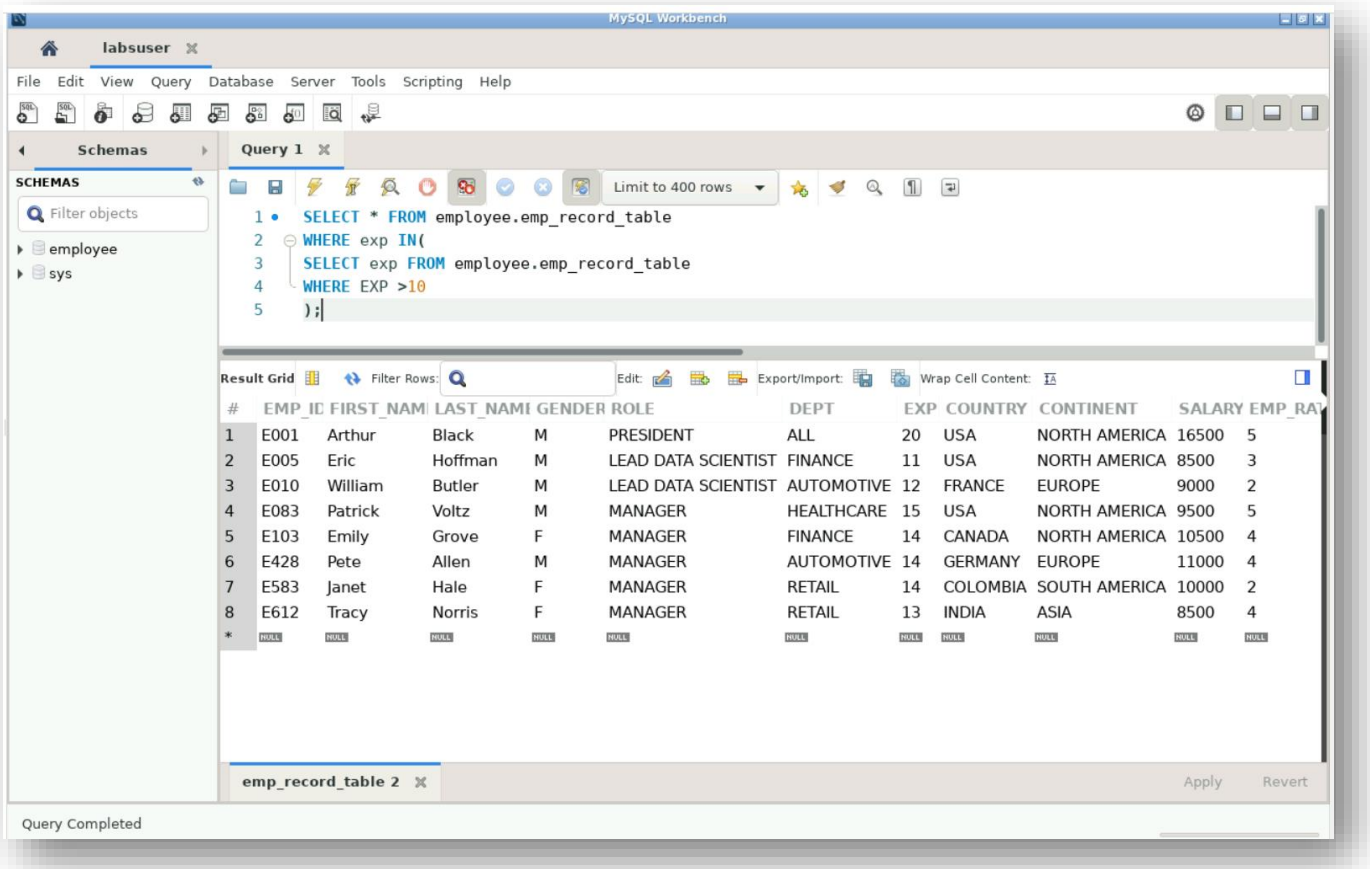


12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

Query-

```
SELECT * FROM employee.emp_record_table
WHERE exp IN (
SELECT exp FROM employee.emp_record_table
WHERE exp > 10
);
```

Output-

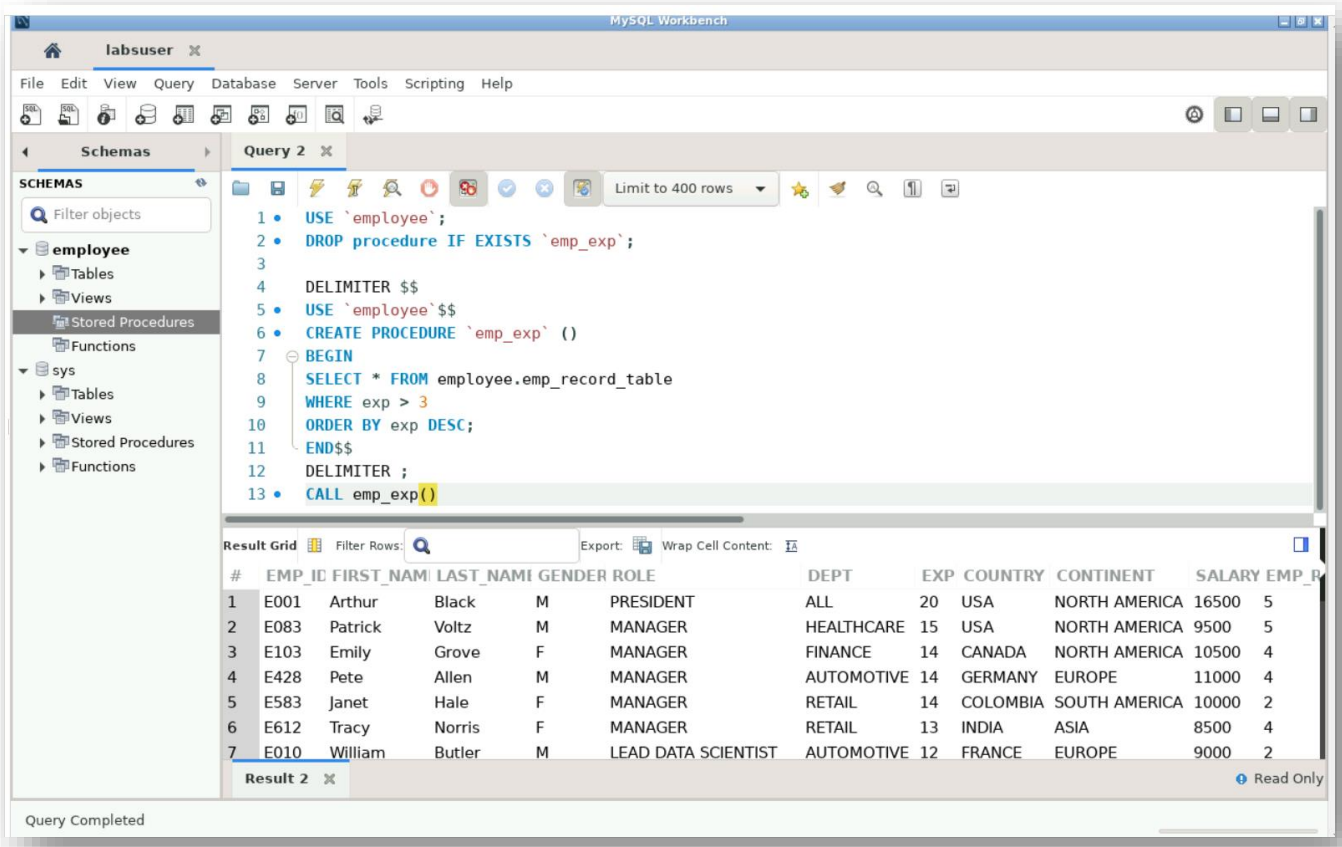


13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.

Query-

```
USE `employee`;
DROP procedure IF EXISTS `emp_exp`;
DELIMITER $$
USE `employee`$$
CREATE PROCEDURE `emp_exp` ()
BEGIN
SELECT * FROM employee.emp_record_table
WHERE exp > 3
ORDER BY exp DESC;
END$$
DELIMITER ;
CALL emp_exp()
```

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:

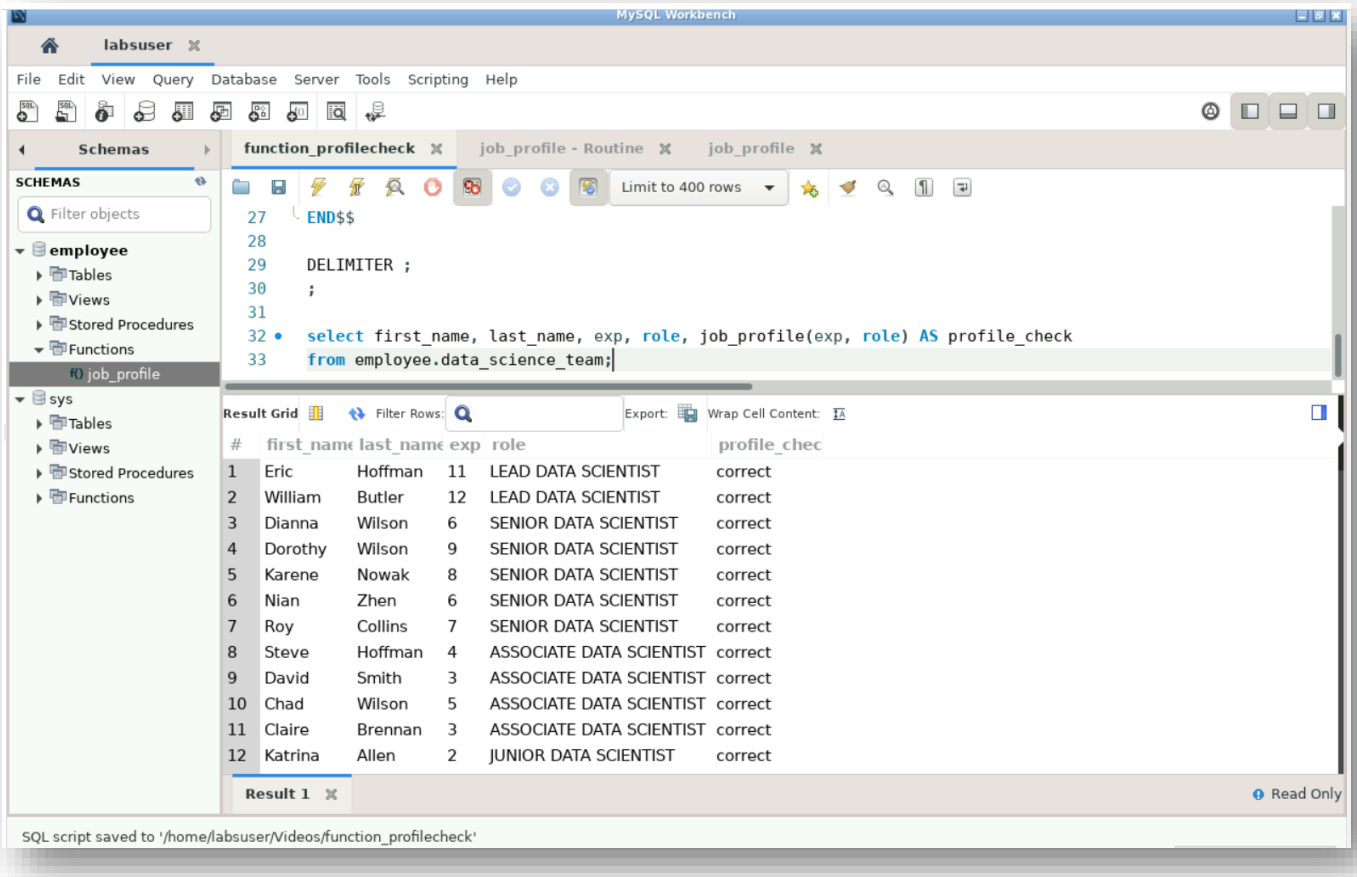
For an employee with experience less than or equal to 2 years assign 'JUNIOR DATA SCIENTIST',
For an employee with the experience of 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST',
For an employee with the experience of 5 to 10 years assign 'SENIOR DATA SCIENTIST',
For an employee with the experience of 10 to 12 years assign 'LEAD DATA SCIENTIST',
For an employee with the experience of 12 to 16 years assign 'MANAGER'.

Query-

```
USE `employee`;
DROP function IF EXISTS `job_profile`;


USE `employee`;
DROP function IF EXISTS `employee`.`job_profile`;
;


DELIMITER $$
USE `employee`$$
CREATE DEFINER=`labsuser`@`localhost` FUNCTION `job_profile`(exp int, role varchar(25))
RETURNS varchar(20) CHARSET utf8mb4
    DETERMINISTIC
BEGIN
DECLARE profile_check varchar(20);
if (exp<=2 AND role="JUNIOR DATA SCIENTIST")
then set profile_check="correct";
elseif(exp>2 AND exp<=5 AND role="ASSOCIATE DATA SCIENTIST")
then set profile_check="correct";
elseif(exp>5 AND exp<=10 AND role="SENIOR DATA SCIENTIST")
then set profile_check="correct";
elseif(exp>10 AND exp<=12 AND role="LEAD DATA SCIENTIST")
then set profile_check="correct";
elseif(exp>12 AND exp<=16 AND role="MANAGER")
then set profile_check="correct";
ELSE set profile_check="not correct";
END IF;
RETURN(profile_check);
```

```
END$$

DELIMITER ;
;

SELECT first_name, last_name, exp, role, job_profile(exp, role) AS profile_check
FROM employee.data_science_team;
```

Output-

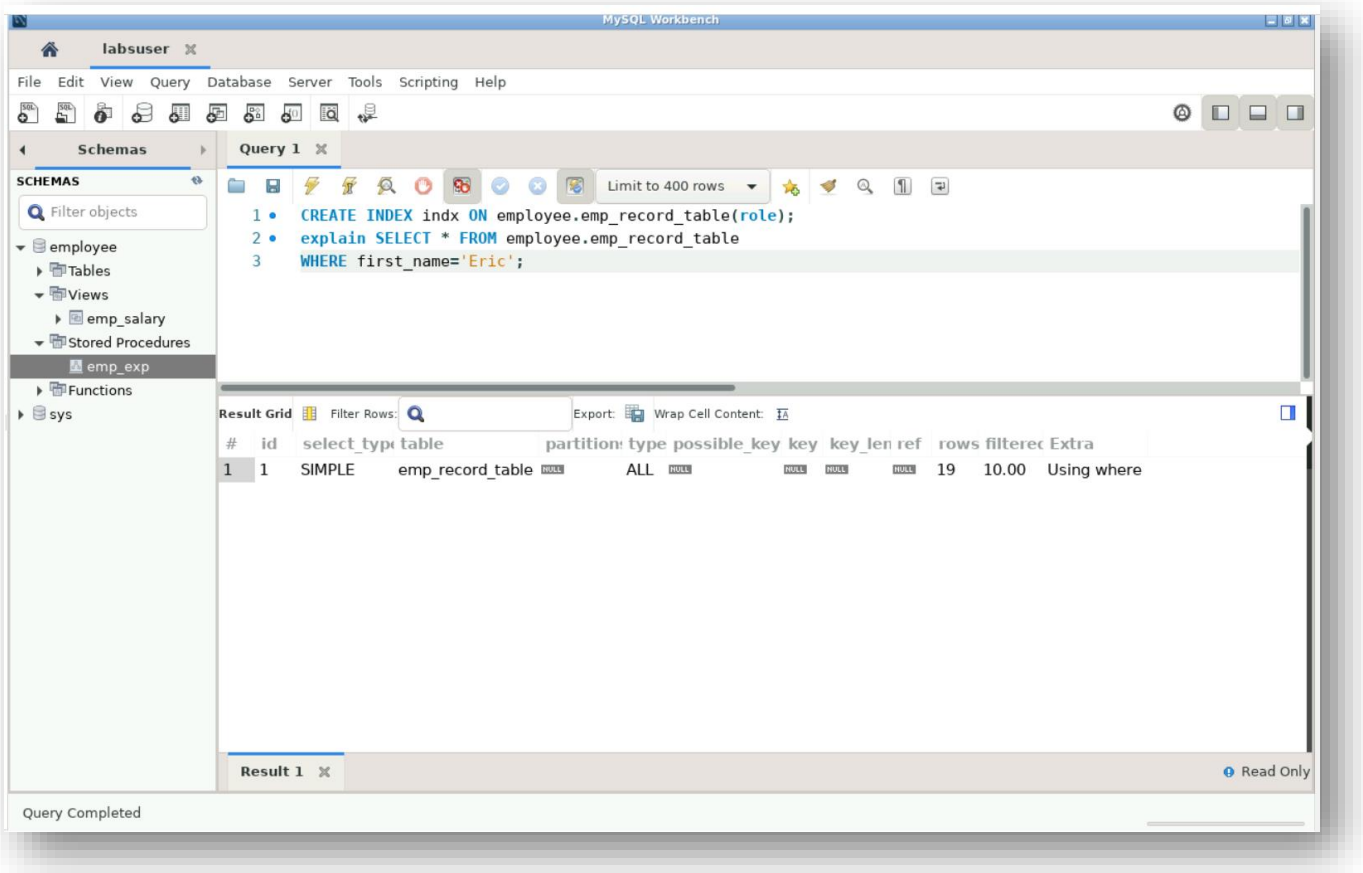


15. Create an index to improve the cost and performance of the query to find the employee whose FIRST_NAME is ‘Eric’ in the employee table after checking the execution plan.

Query-

```
CREATE INDEX indx ON employee.emp_record_table(role);
EXPLAIN SELECT * FROM employee.emp_record_table
WHERE first_name='Eric';
```

Output-

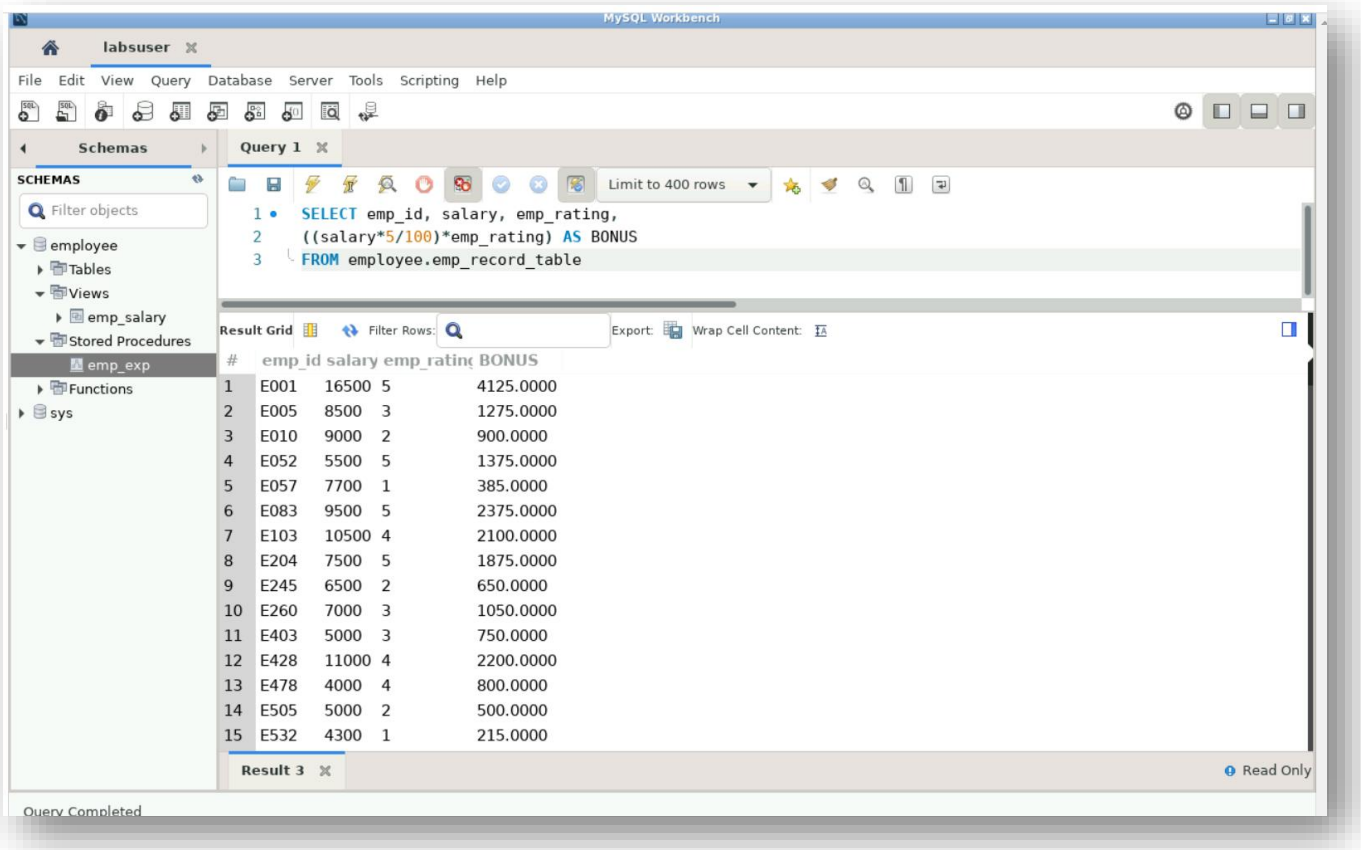


16. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary * employee rating).

Query-

```
SELECT  emp_id, salary, emp_rating,
((salary*5/100)*emp_rating) AS bonus
FROM employee.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  emp_id, salary, emp_rating,
SELECT continent, avg(salary) FROM employee.emp_record_table GROUP BY continent;
SELECT country, avg(salary) FROM employee.emp_record_table GROUP BY country;
SELECT continent,
(SELECT avg(y.salary) FROM employee.emp_record_table y
WHERE x.continent = y.continent) AS continent_avg, country,
(SELECT avg(z.salary) FROM employee.emp_record_table z
WHERE x.continent = z.continent) AS country_avg
FROM employee.emp_record_table x
GROUP BY continent, country;
```

Output-