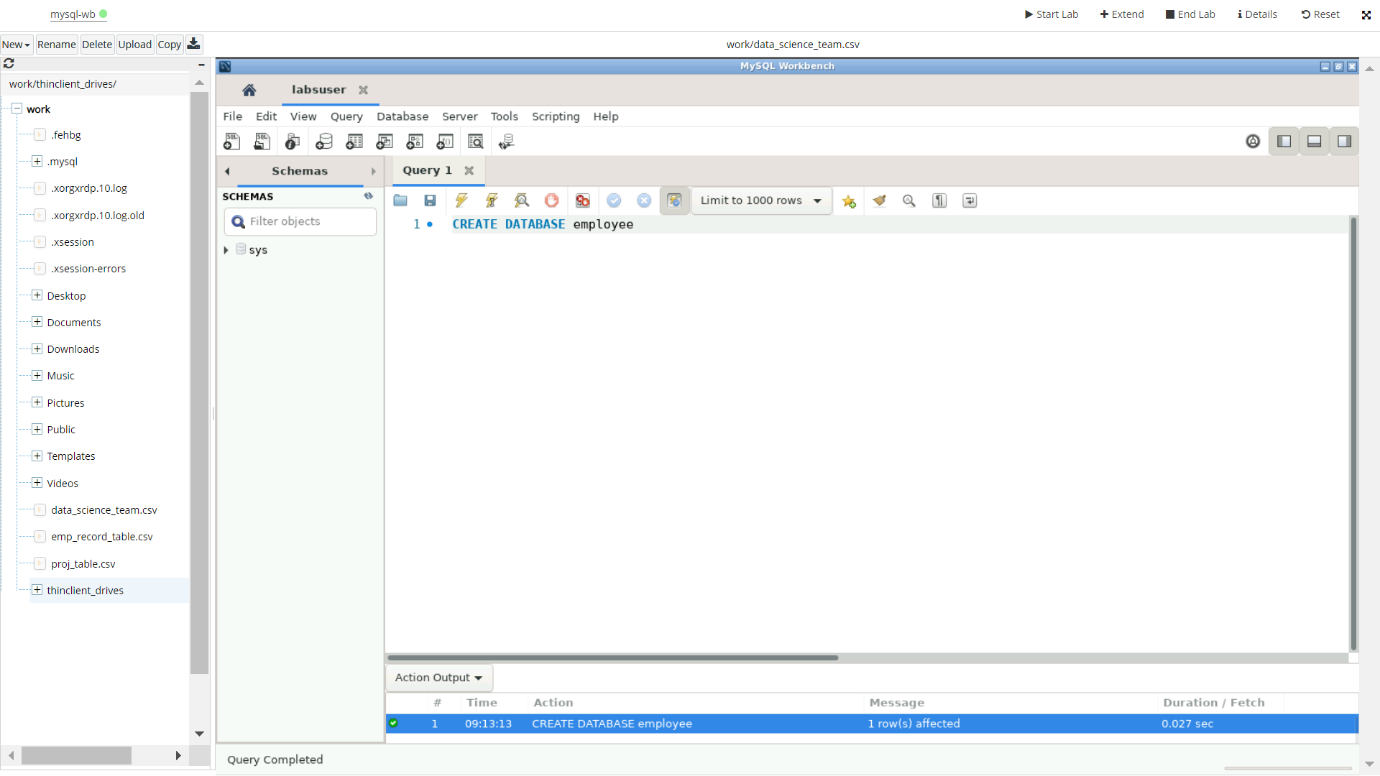
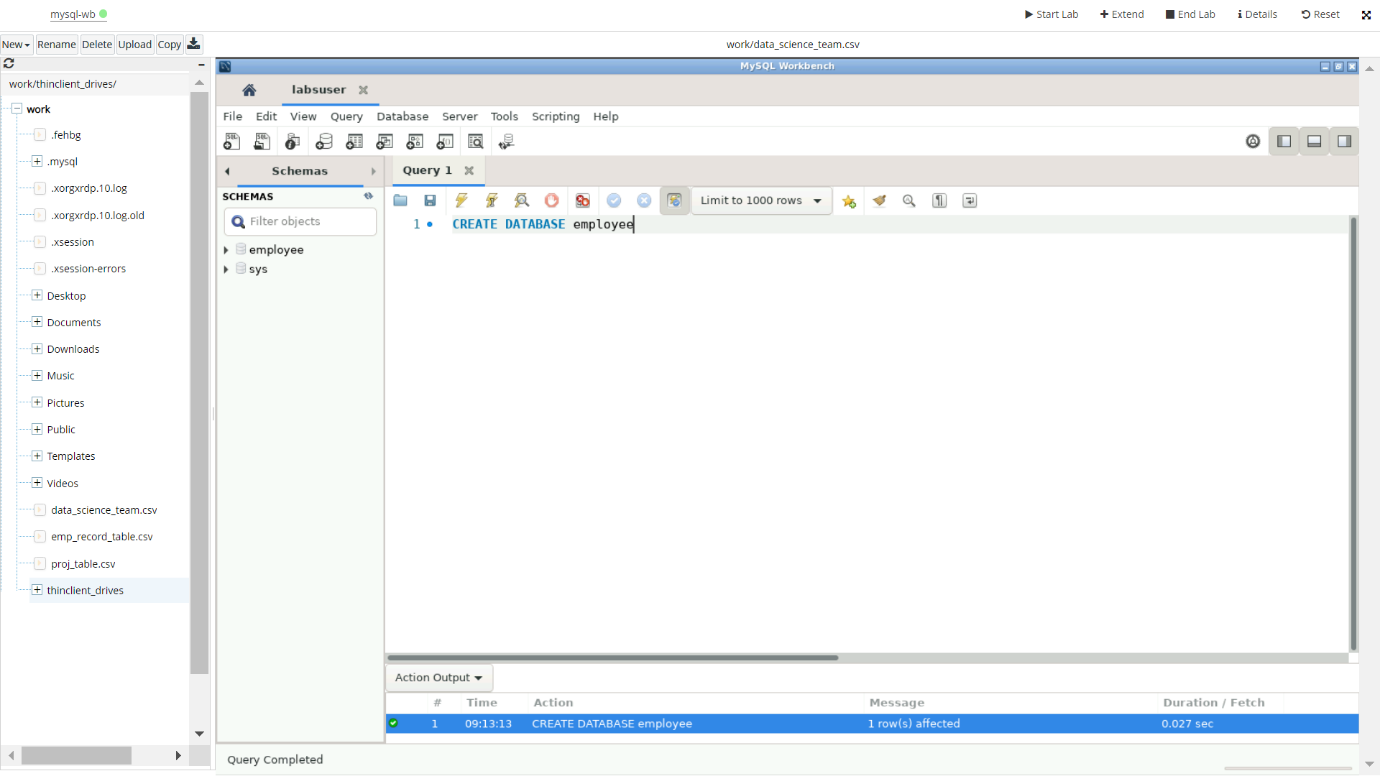
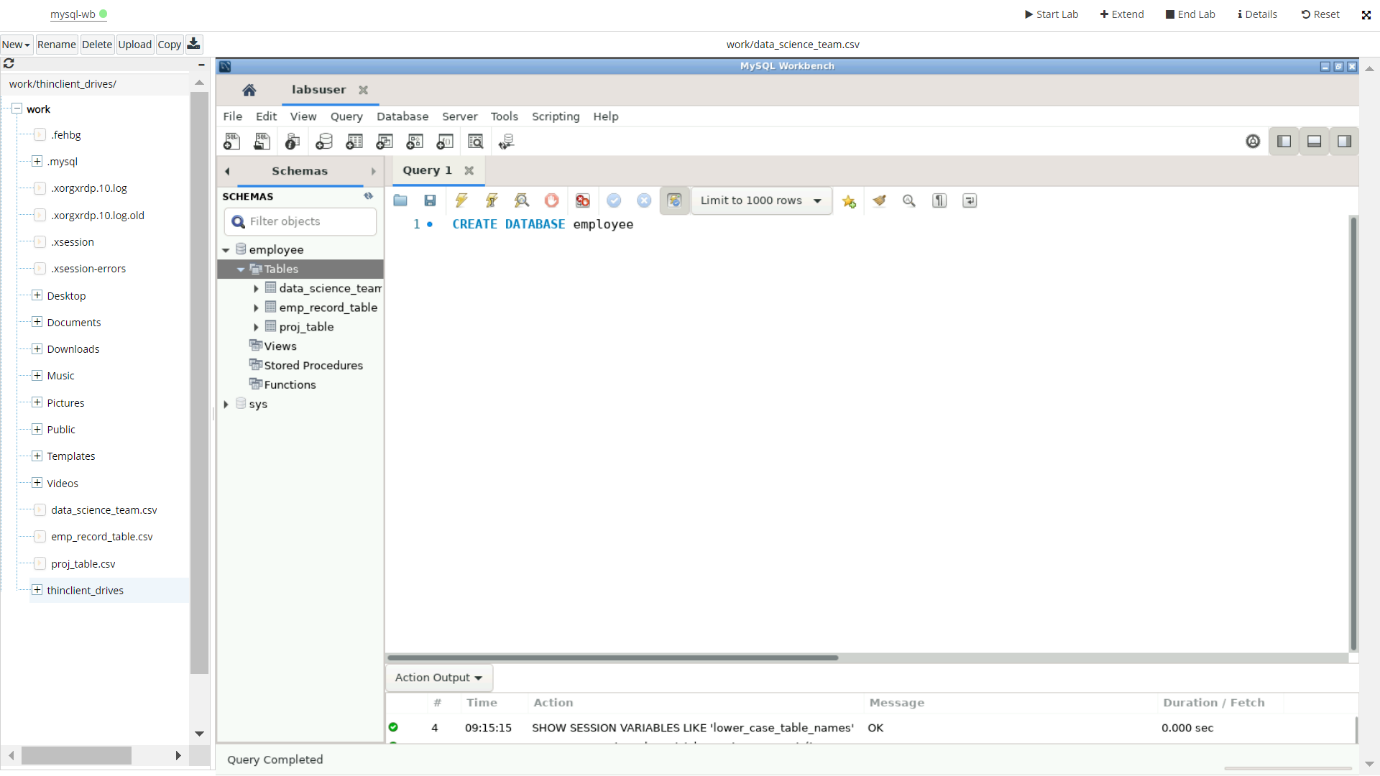
**The task to be performed:**

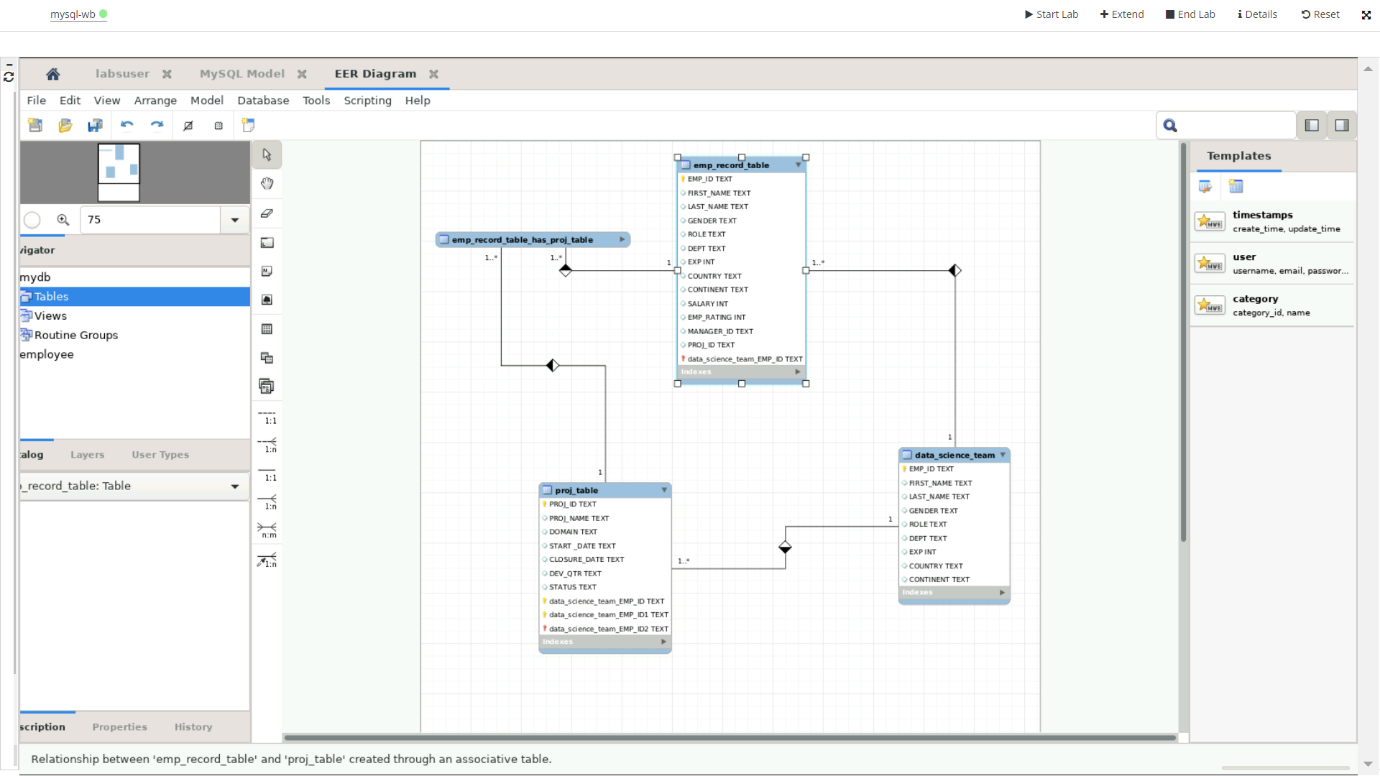
1. **Create a database named employee, then import data\_science\_team.csv proj\_table.csv and emp\_record\_table.csv into the employee database from the given resources.**







**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.**

**Query to fetch details from employee record table:**

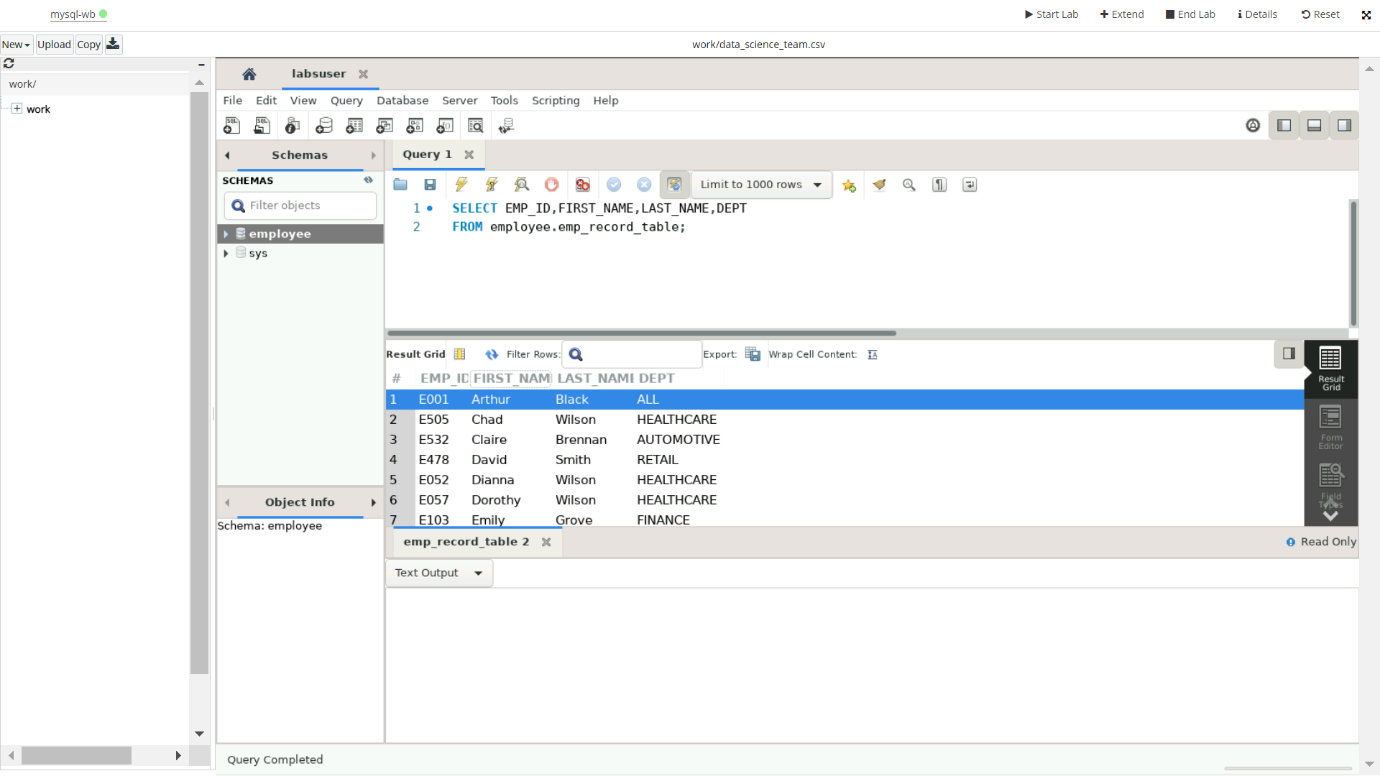
SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT

FROM employee.emp\_record\_table;

**List of employees and details of their department:**

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

FROM employee.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:**

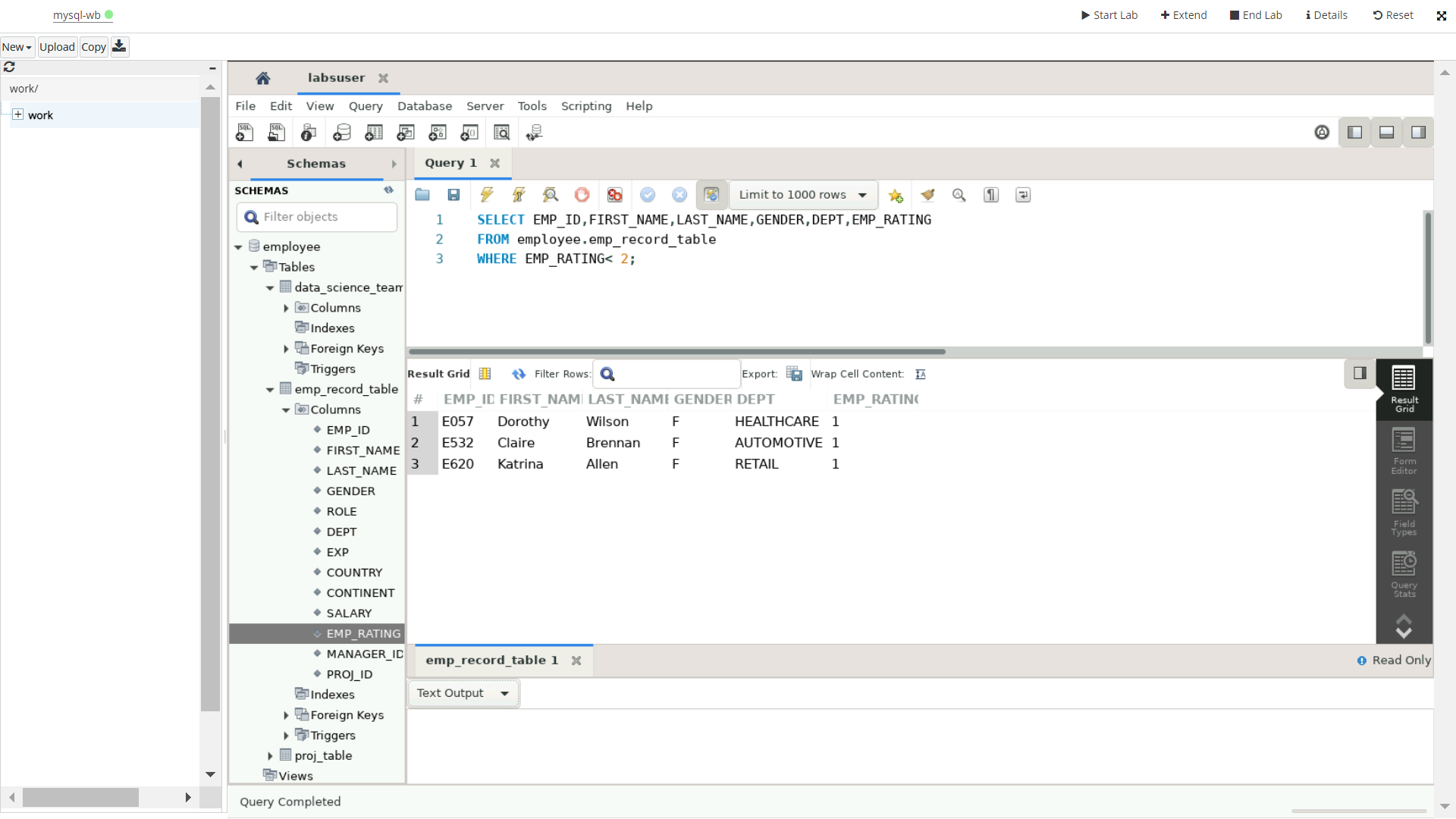
* **less than two**

Query:

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

FROM employee.emp\_record\_table

WHERE EMP\_RATING<2;



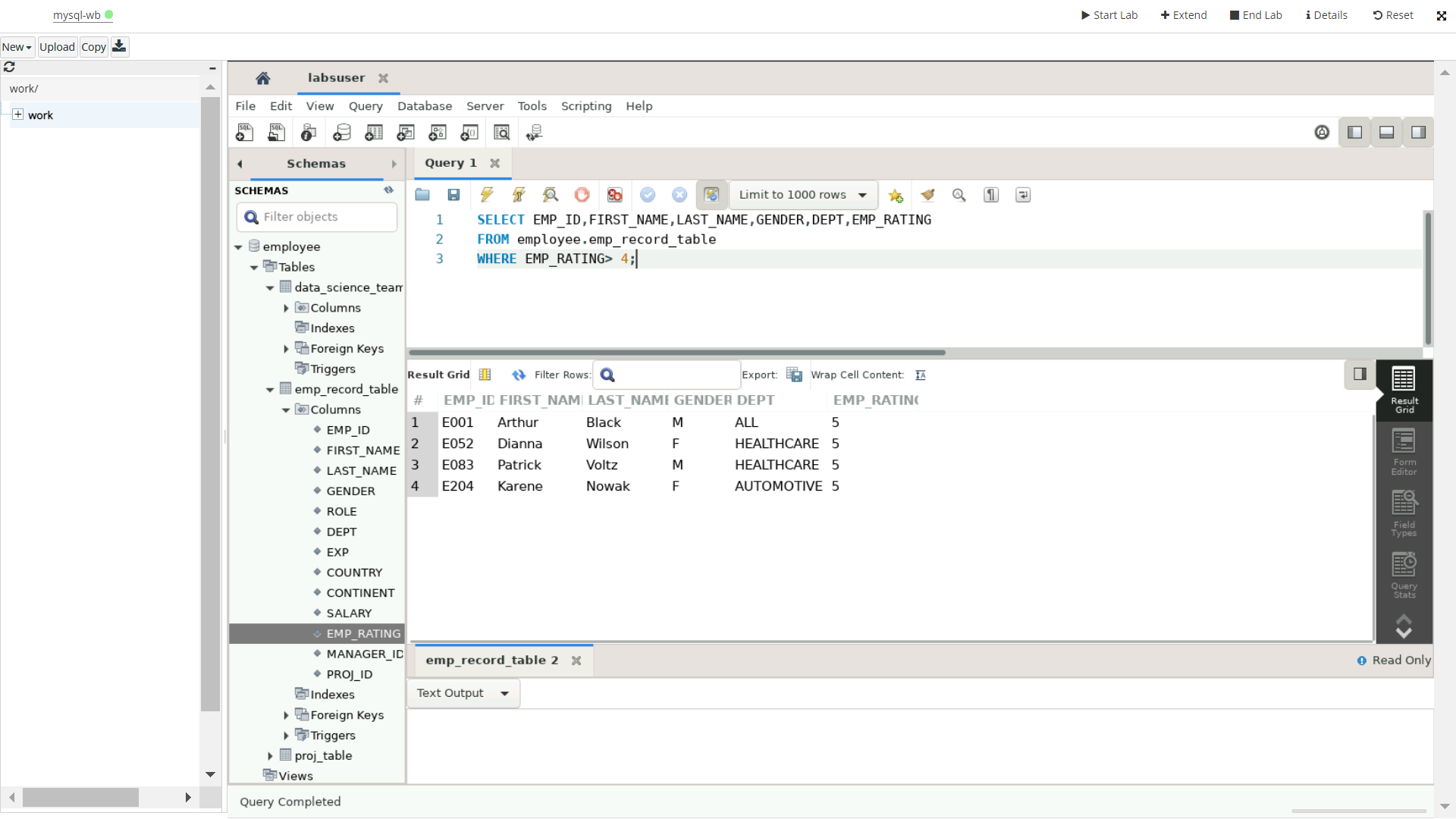
* **greater than four**

Query:

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

FROM employee.emp\_record\_table

WHERE EMP\_RATING>4;



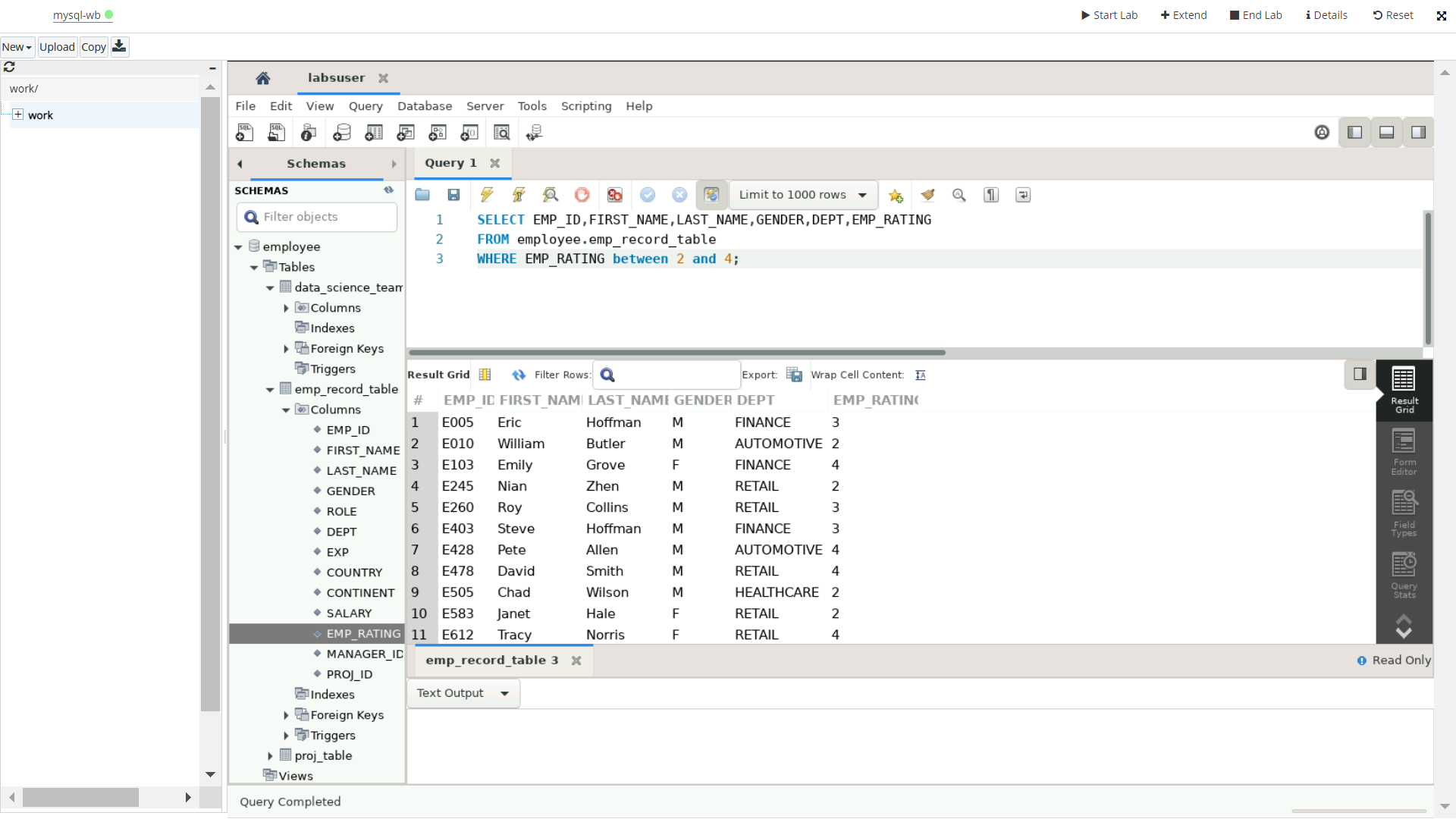
* **between two and four**

Query:

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

FROM employee.emp\_record\_table

WHERE EMP\_RATING BETWEEN 2 AND 4;



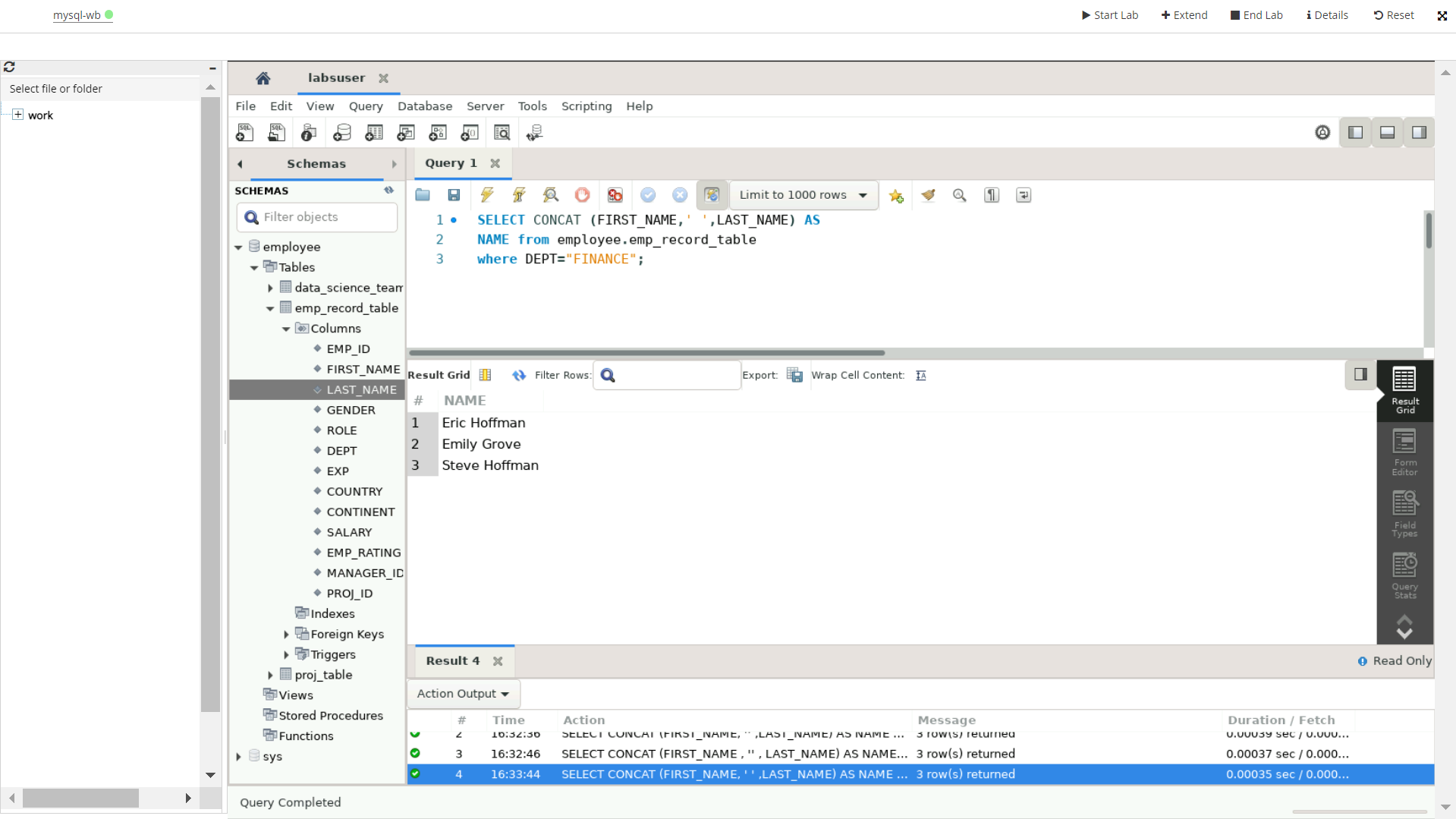
**5.Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.**

QUERY:

SELECT CONCAT (FIRST\_NAME,’ ‘, LAST\_NAME) AS

NAME FROM employee.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).**

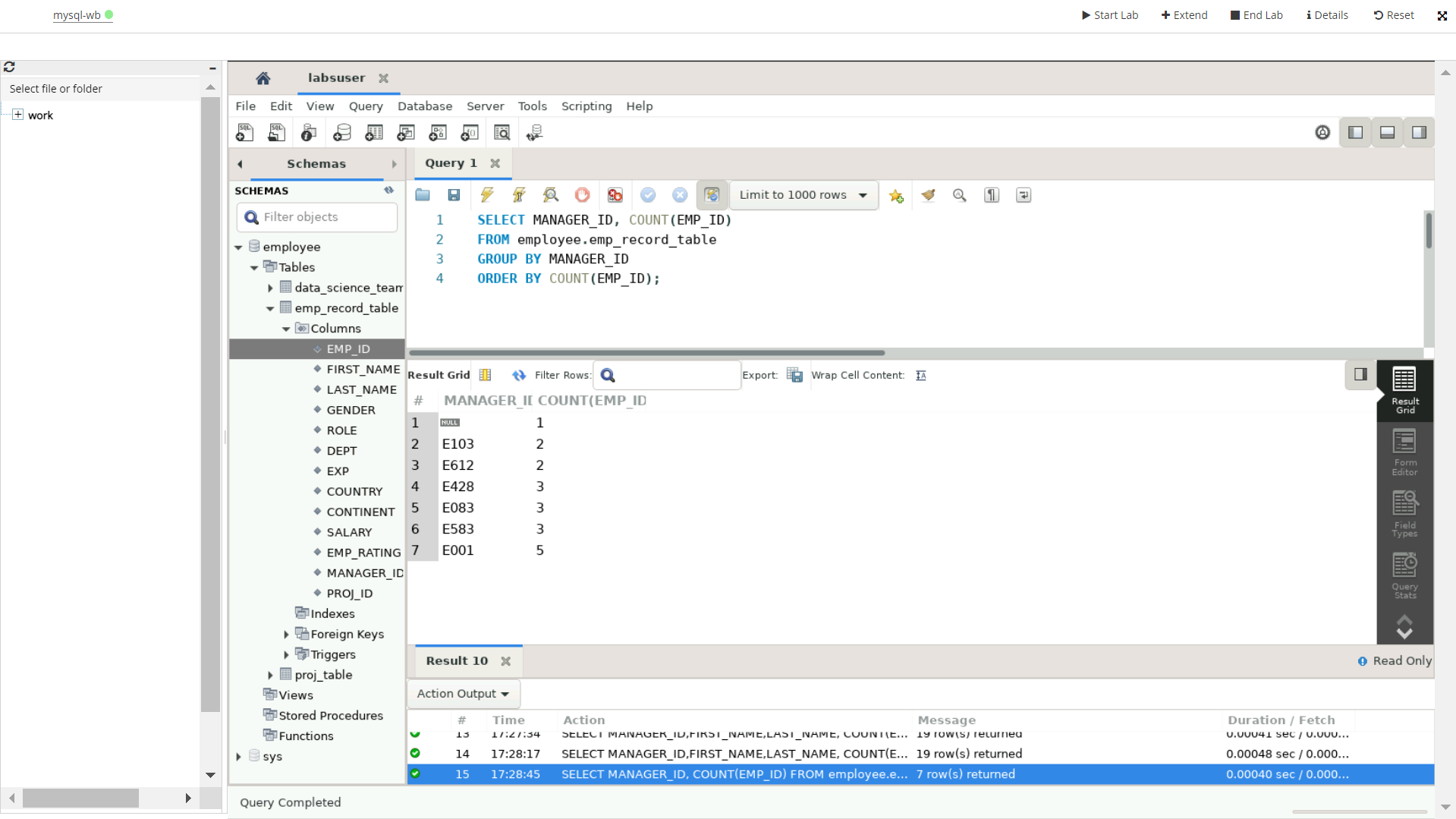
QUERY:

SELECT MANAGER\_ID, COUNT(EMP\_ID)

FROM employee.emp\_record\_table

GROUP BY MANAGER\_ID

ORDER BY COUNT(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.**

QUERY:

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

FROM employee.emp\_record\_table

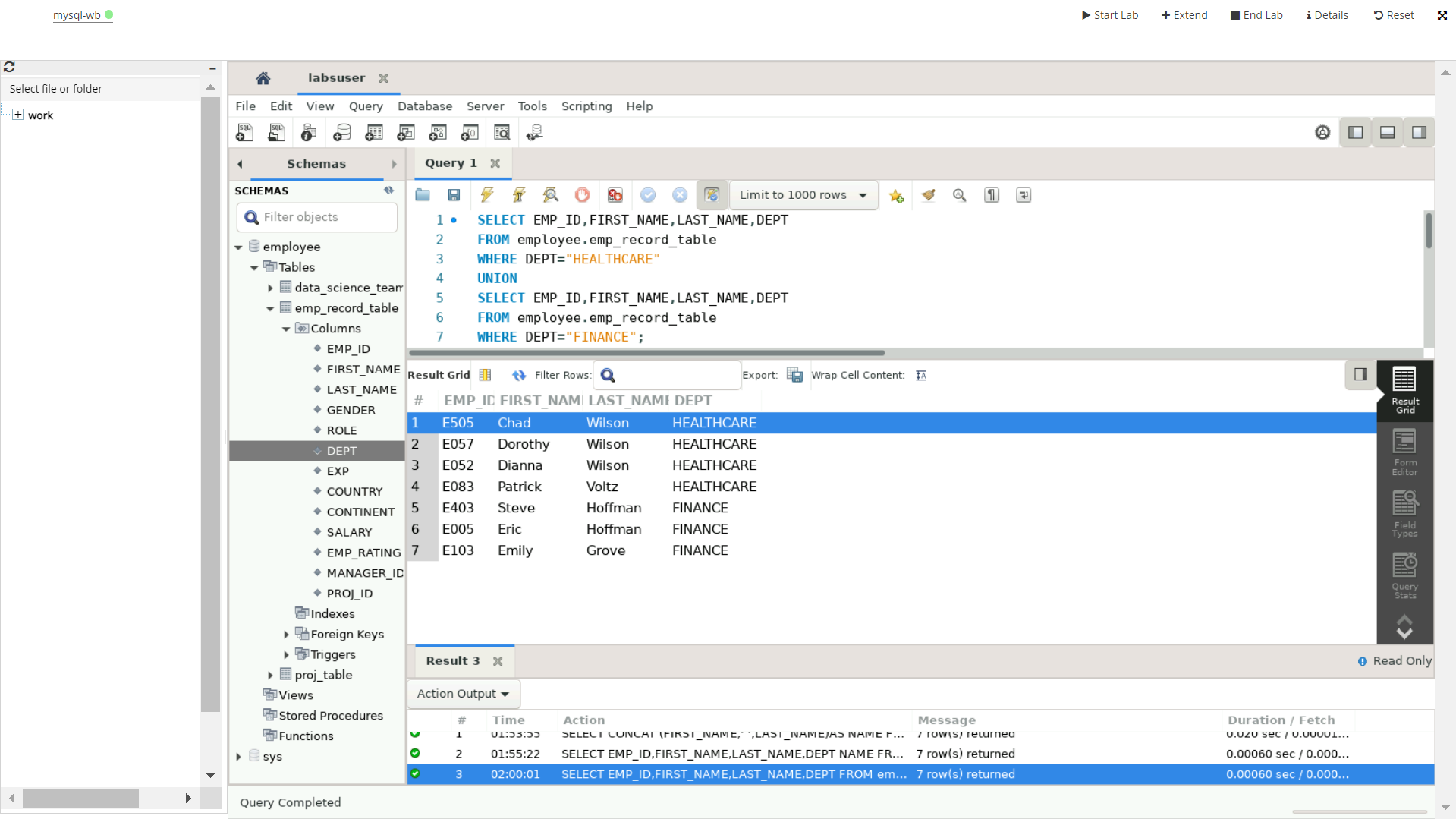
WHERE DEPT=”HEALTHCARE”

UNION

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

FROM employee.emp\_record\_table

WHERE DEPT=”FINANCE”;



**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, ROLE, DEPT, EMP\_RATING

FROM employee.emp\_record\_table

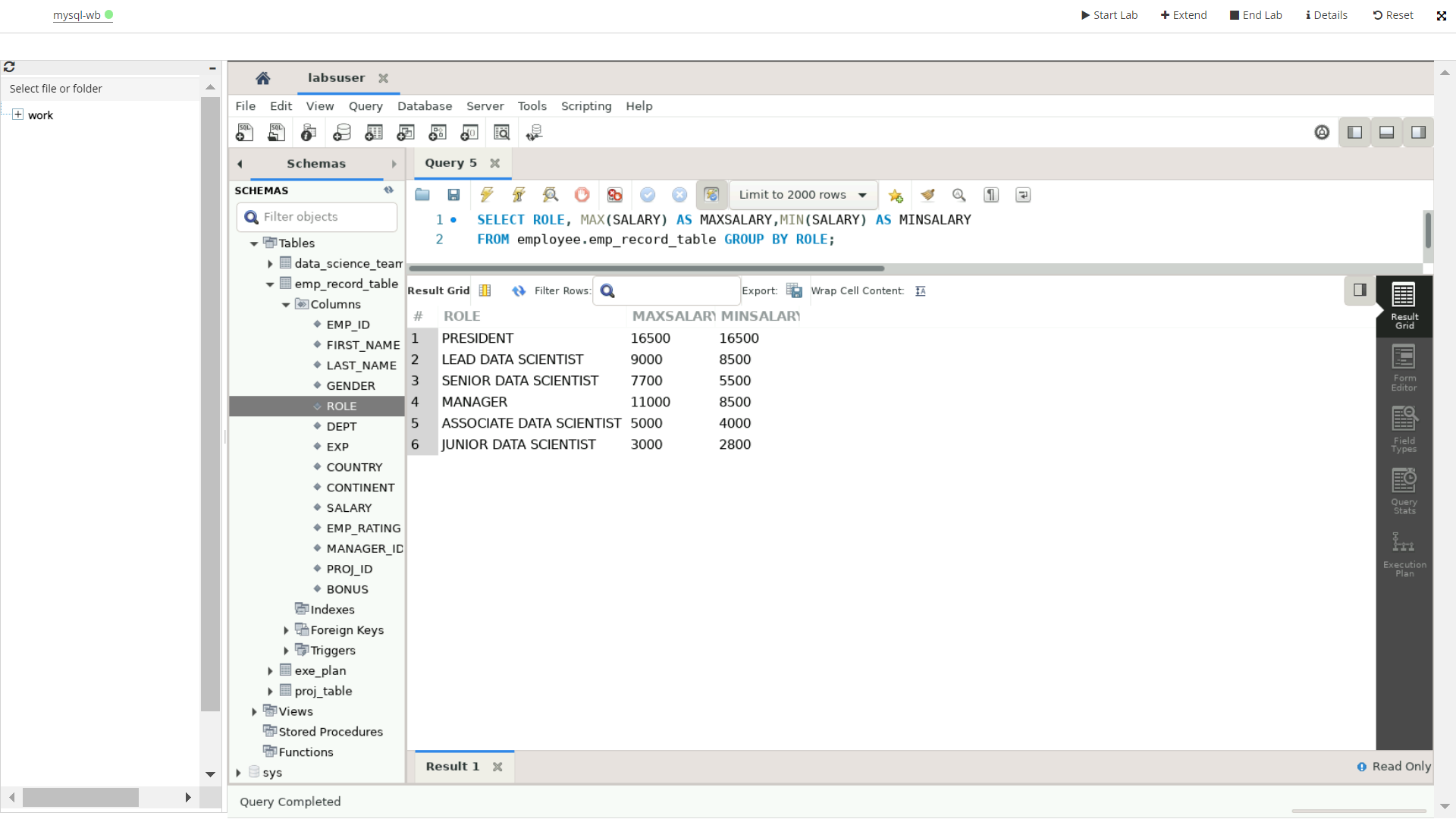
WHERE (EMP\_RATING,DEPT) IN (SELECT MAX(EMP\_RATING),DEPT FROM employee.emp\_record\_table GROUP BY DEPT);

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

QUERY:

SELECT ROLE, MAX(SALARY) AS MAXSALARY, MIN(SALARY) AS MINSALARY

FROM employee.emp\_record\_table GROUP BY ROLE;



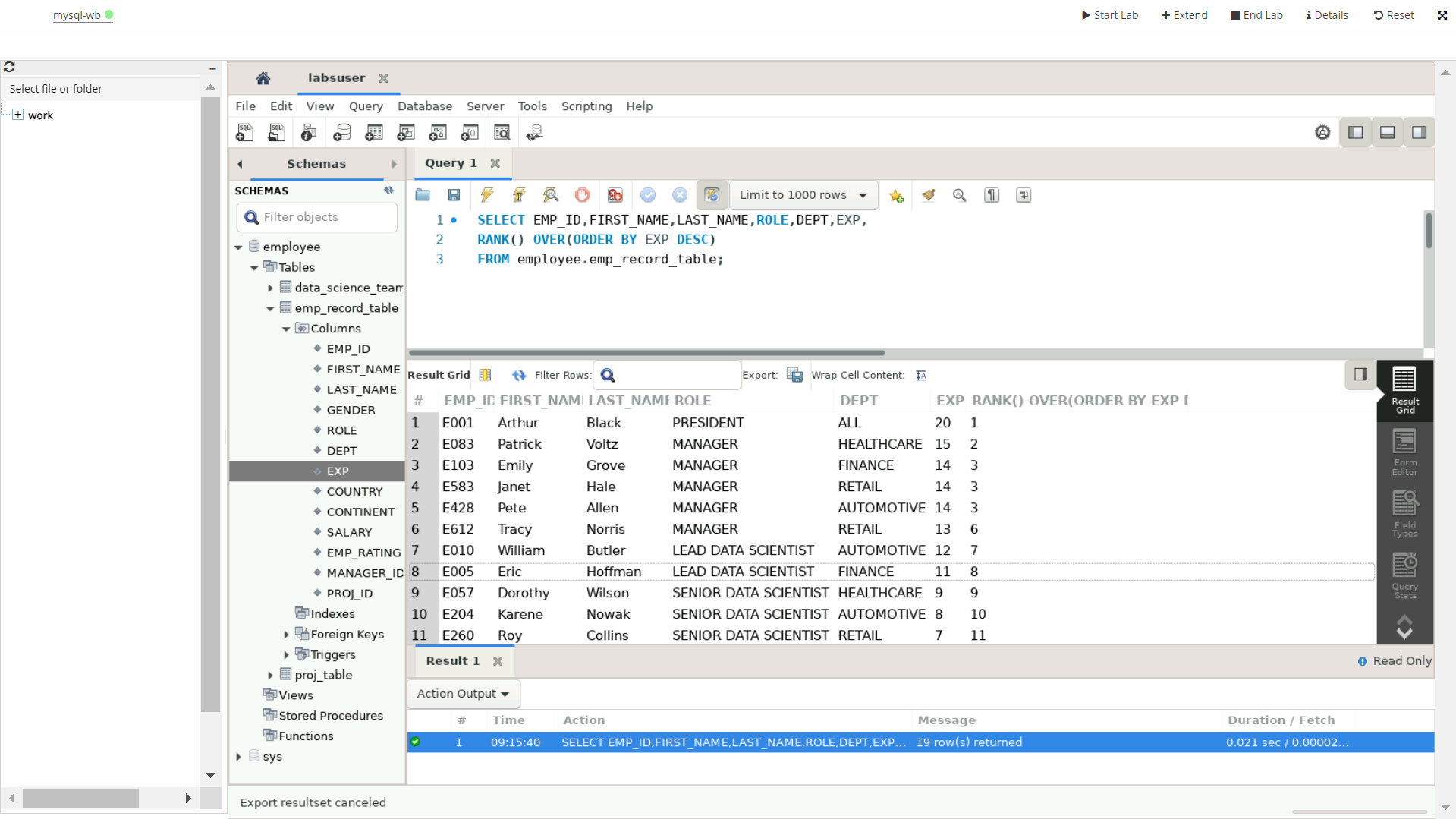
**10.Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.**

QUERY:

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

RANK () OVER (ORDER BY EXP DESC)

FROM employee.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.**

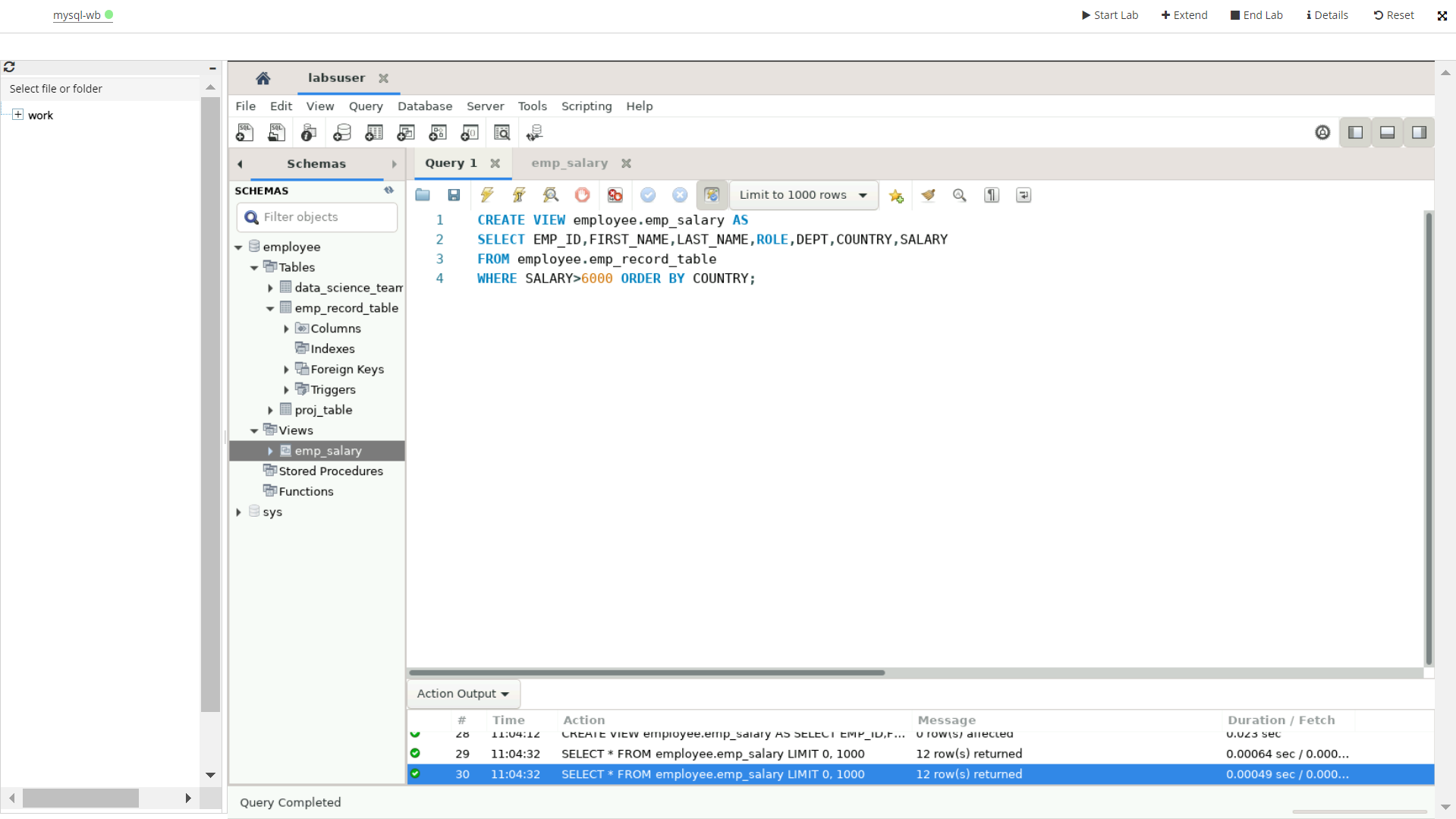
QUERY- To create a view

CREATE VIEW employee.emp\_salary AS

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

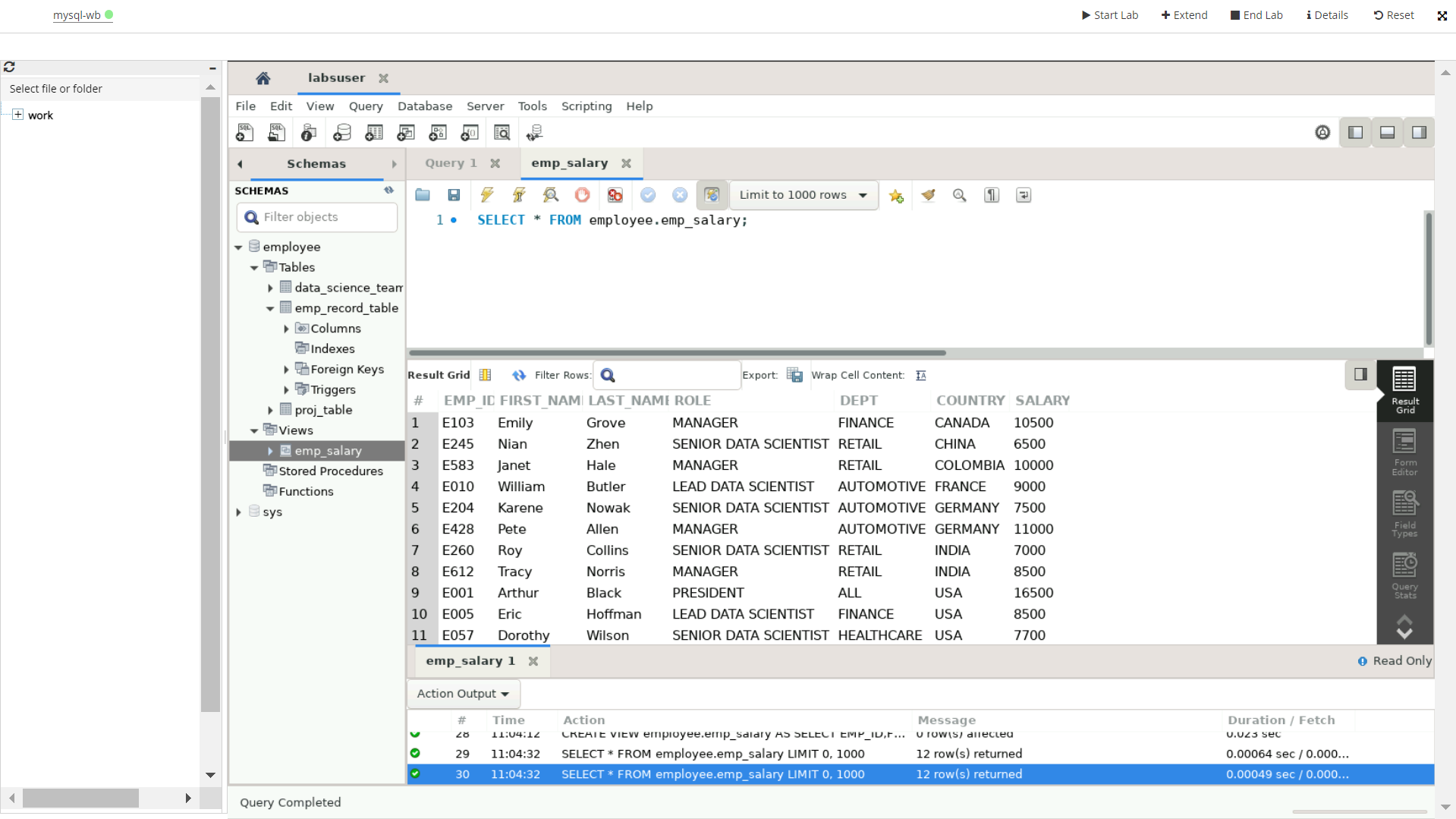
FROM employee.emp\_record\_table

WHERE SALARY>6000 ORDER BY COUNTRY;



QUERY- To see the view contents:

SELECT \* FROM employee.emp\_salary;



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

QUERY:

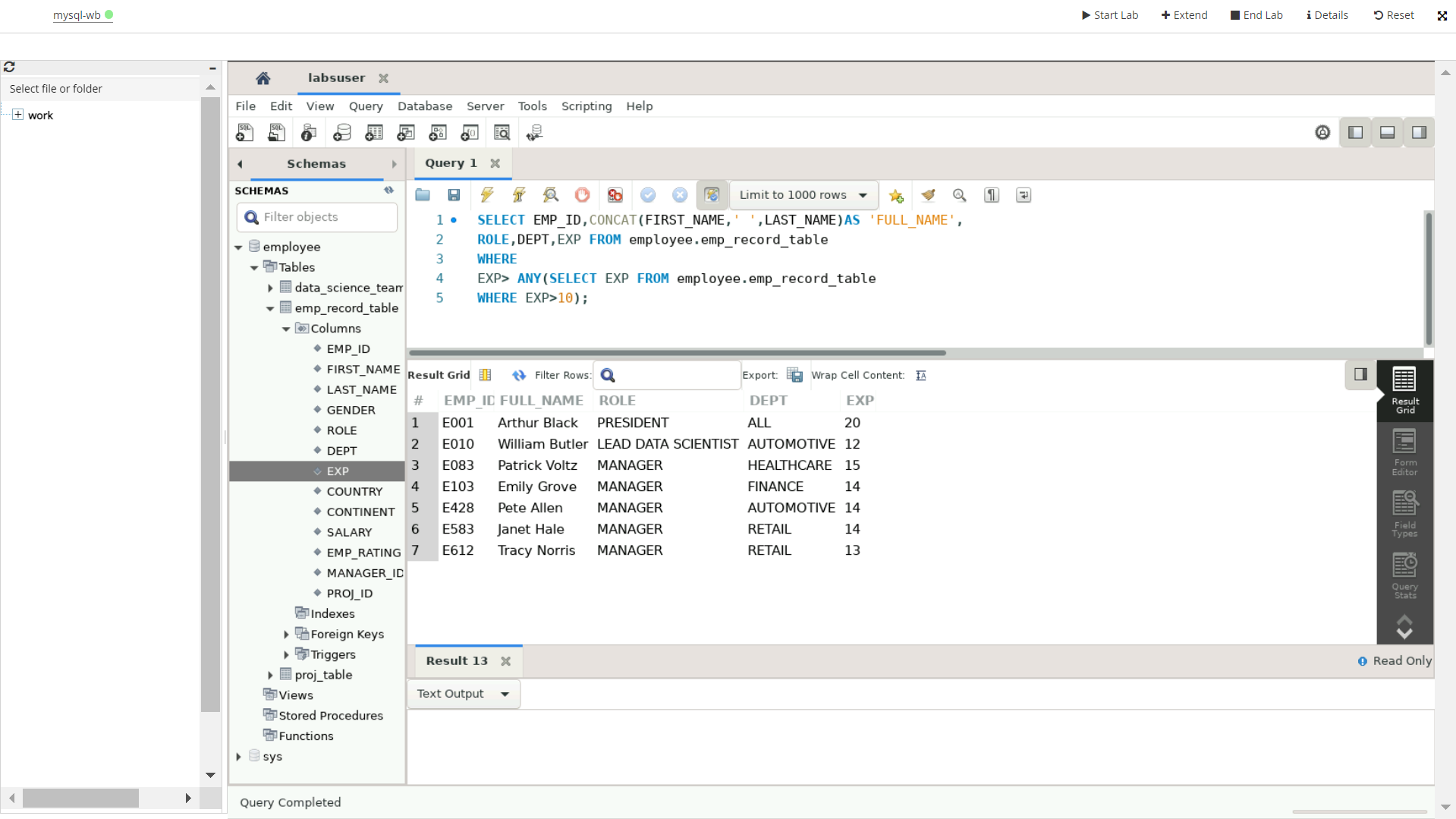
SELECT EMP\_ID, CONCAT (FIRST\_NAME,’ ‘, LAST\_NAME) AS ‘FULL\_NAME’,

ROLE, DEPT, EXP FROM employee.emp\_record\_table

WHERE

EXP>ANY (SELECT EXP FROM employee.emp\_record\_table

WHERE EXP>10);

****

**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:

To create a stored procedure-

DELIMITER &&

CREATE PROCEDURE get\_experience ()

BEGIN

SELECT \* FROM employee.emp\_record\_table

WHERE EXP>3;

END &&

To call the for the stored procedure-

CALL get\_experience();

**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'.**

DELIMITER $$

drop function employee.role;

CREATE FUNCTION role (exp int)

RETURNS VARCHAR (2255) DETERMINISTIC

BEGIN DECLARE role VARCHAR (2255);

IF experience <= 2 THEN SET role = 'JUNIOR DATA SCIENTIST';

ELSEIF experience <= 5 THEN SET role = 'ASSOCIATE DATA SCIENTIST';

ELSEIF experience <= 10 THEN SET role = 'SENIOR DATA SCIENTIST';

ELSEIF experience <= 12 THEN SET role = 'LEAD DATA SCIENTIST';

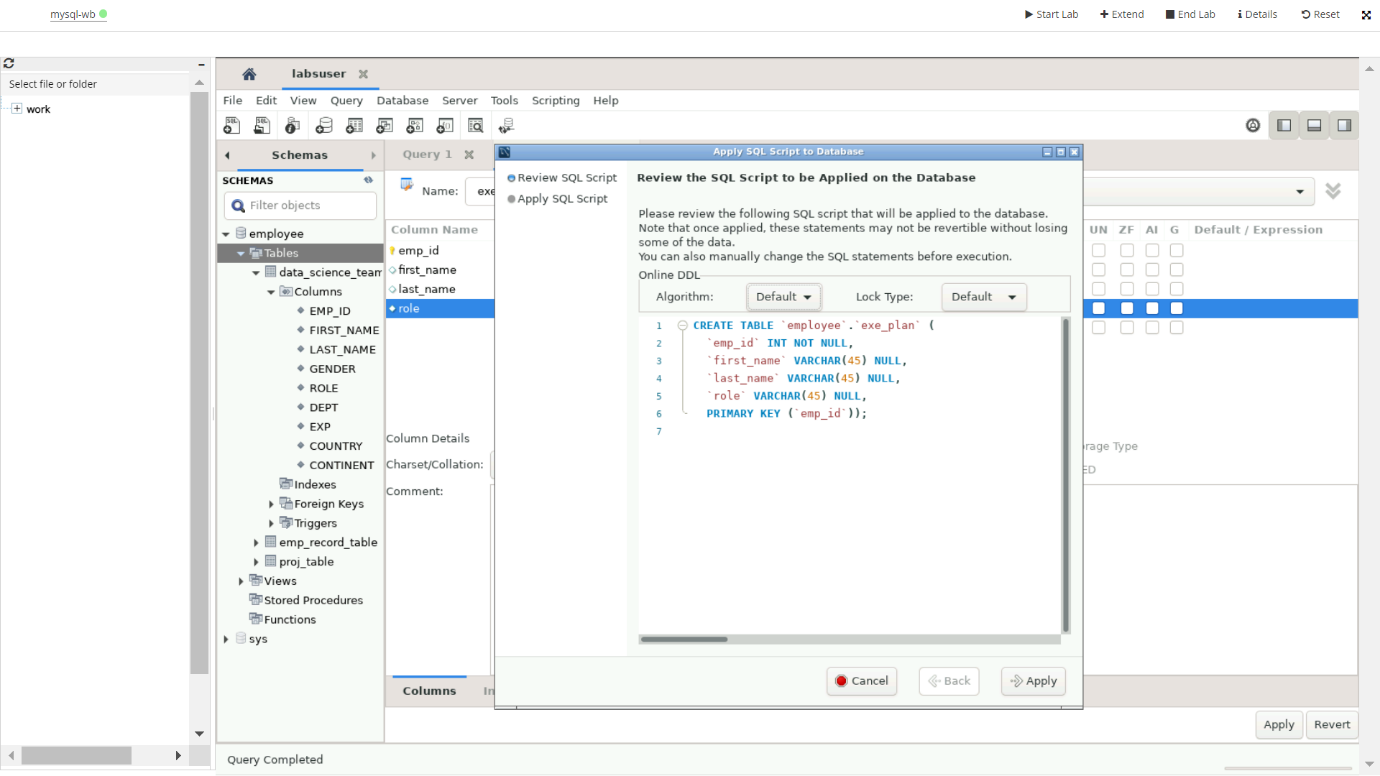
ELSEIF experience > 12 THEN SET role = 'MANAGER'

END IF; RETURN (role); END$$ DELIMITER $$;

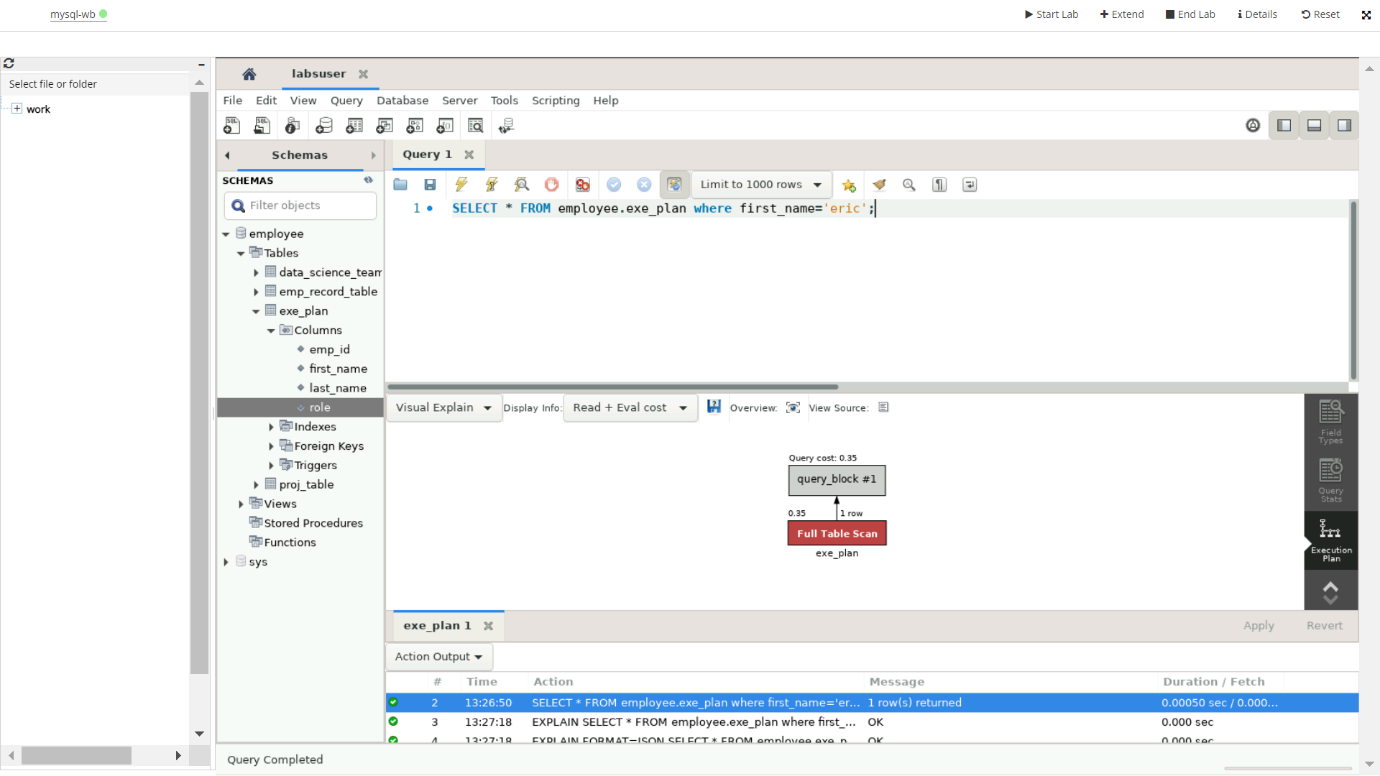
SELECT first\_name, last\_name, dept, Role(exp) as designation FROM employee.data\_science\_team ORDER BY exp;

**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.**

CREATION OF TABLE-



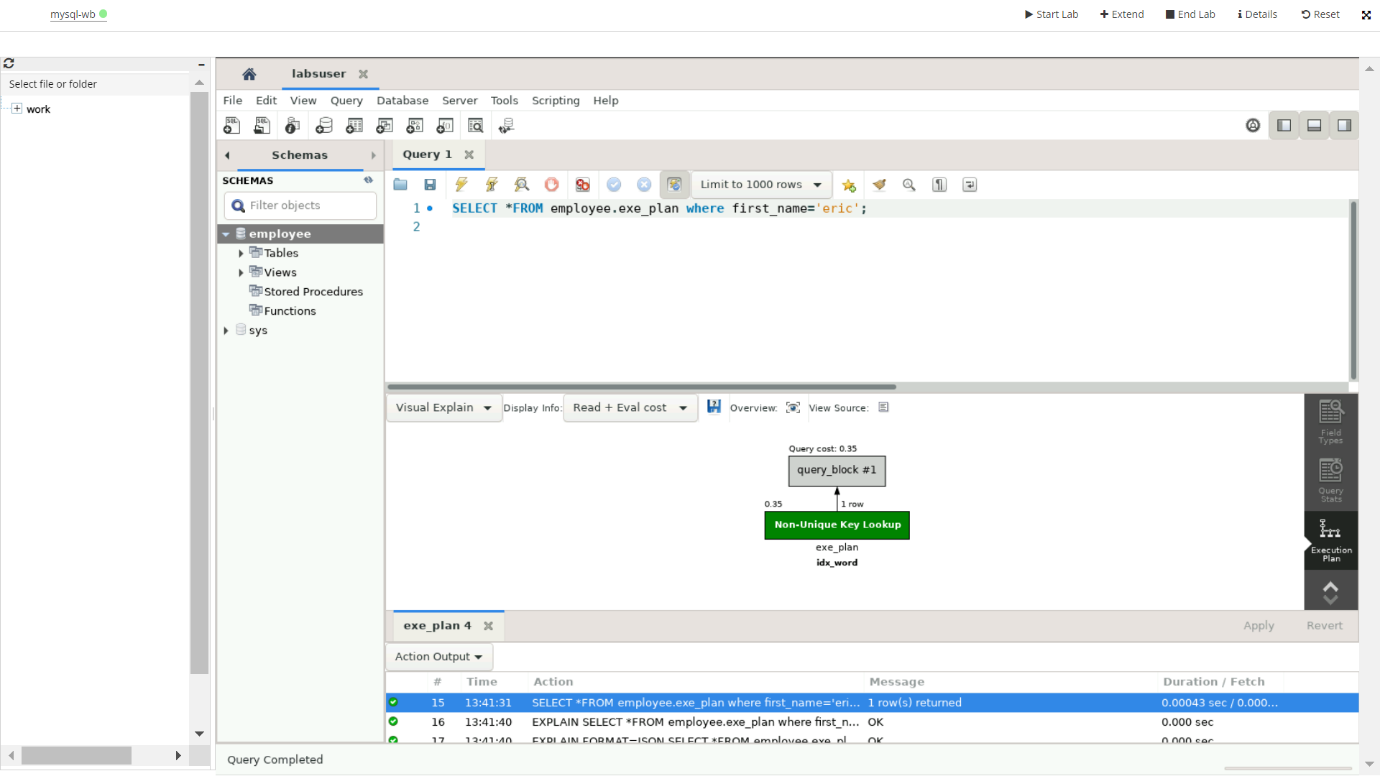
Query cost and performance before creating index-



Query cost and performance after creating index-

Index created by-

CREATE INDEX index\_word on employee.exe\_plan (FIRST\_NAME);



**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:

ALTER TABLE employee.emp\_record\_table

ADD COLUMN BONUS DOUBLE

GENERATED ALWAYS AS (5%(SALARY)\*EMP\_RATING) STORED;

SELECT \* FROM employee.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.**

QUERY-

SELECT COUNTRY, CONTINENT, AVG(SALARY) AS AVERAGE\_SALARY

FROM employee.emp\_record\_table GROUP BY COUNTRY, CONTINENT;

