SOURCECODE SECTION

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;

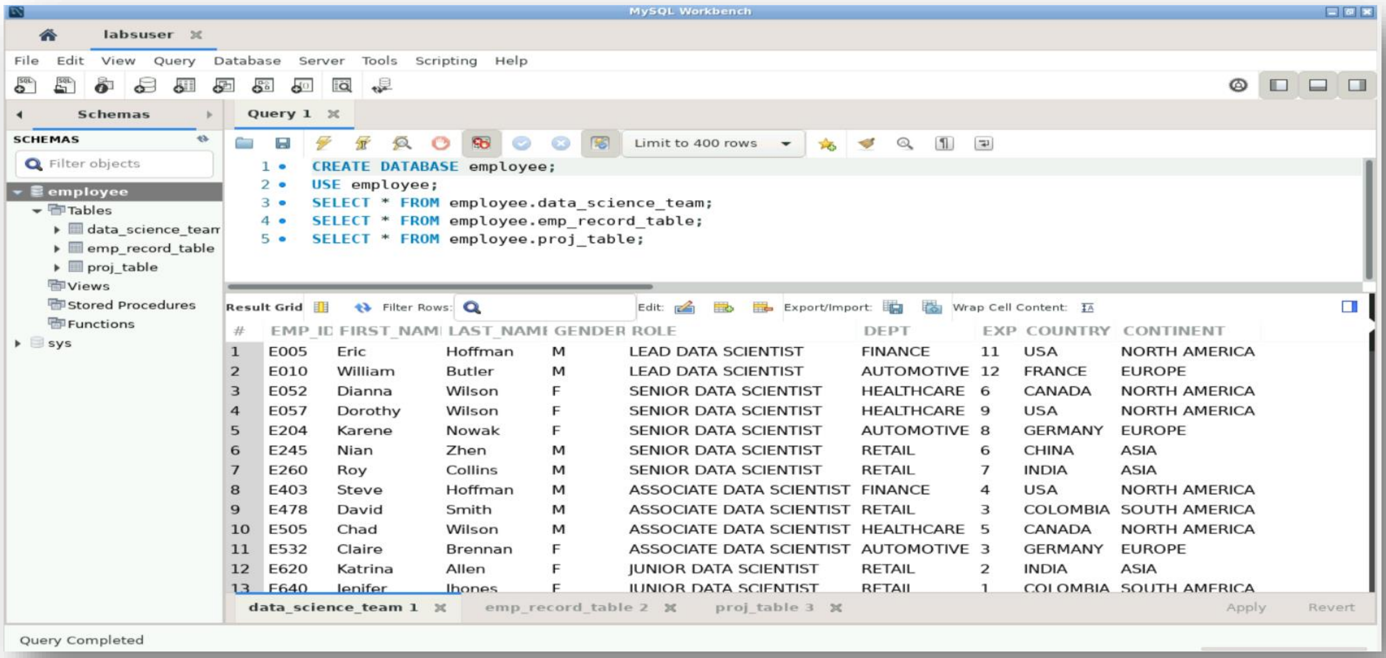
- **Imported emp_record_table.csv , data_science_team.csv and proj_table.csv using Table Data Import Wizard option.**

SELECT *FROM emp_record_table;

SELECT *FROM data_science_team;

SELECT *FROM portable;

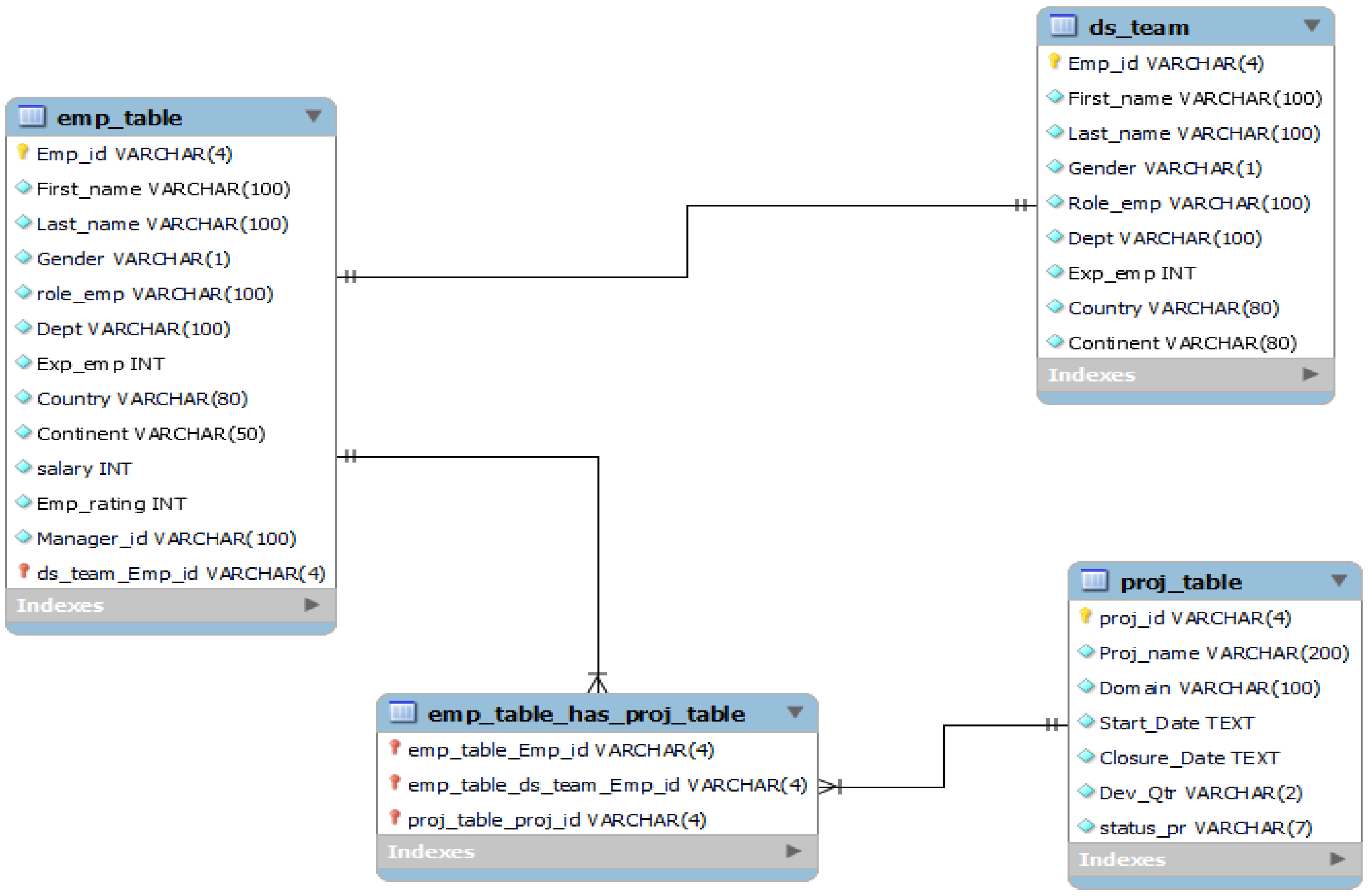
Output-



```
CREATE DATABASE employee;
USE employee;
SELECT * FROM employee.data_science_team;
SELECT * FROM employee.emp_record_table;
SELECT * FROM employee.proj_table;
```

#	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT
1	E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA
2	E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE
3	E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA
4	E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA
5	E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE
6	E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA
7	E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA
8	E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA
9	E478	David	Smith	M	ASSOCIATE DATA SCIENTIST	RETAIL	3	COLOMBIA	SOUTH AMERICA
10	E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA
11	E532	Claire	Brennan	F	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	3	GERMANY	EUROPE
12	E620	Katrina	Allen	F	JUNIOR DATA SCIENTIST	RETAIL	2	INDIA	ASIA
13	E640	Jenifer	Jhones	F	JUNIOR DATA SCIENTIST	RETAIL	1	COLOMBIA	SOUTH AMERICA

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

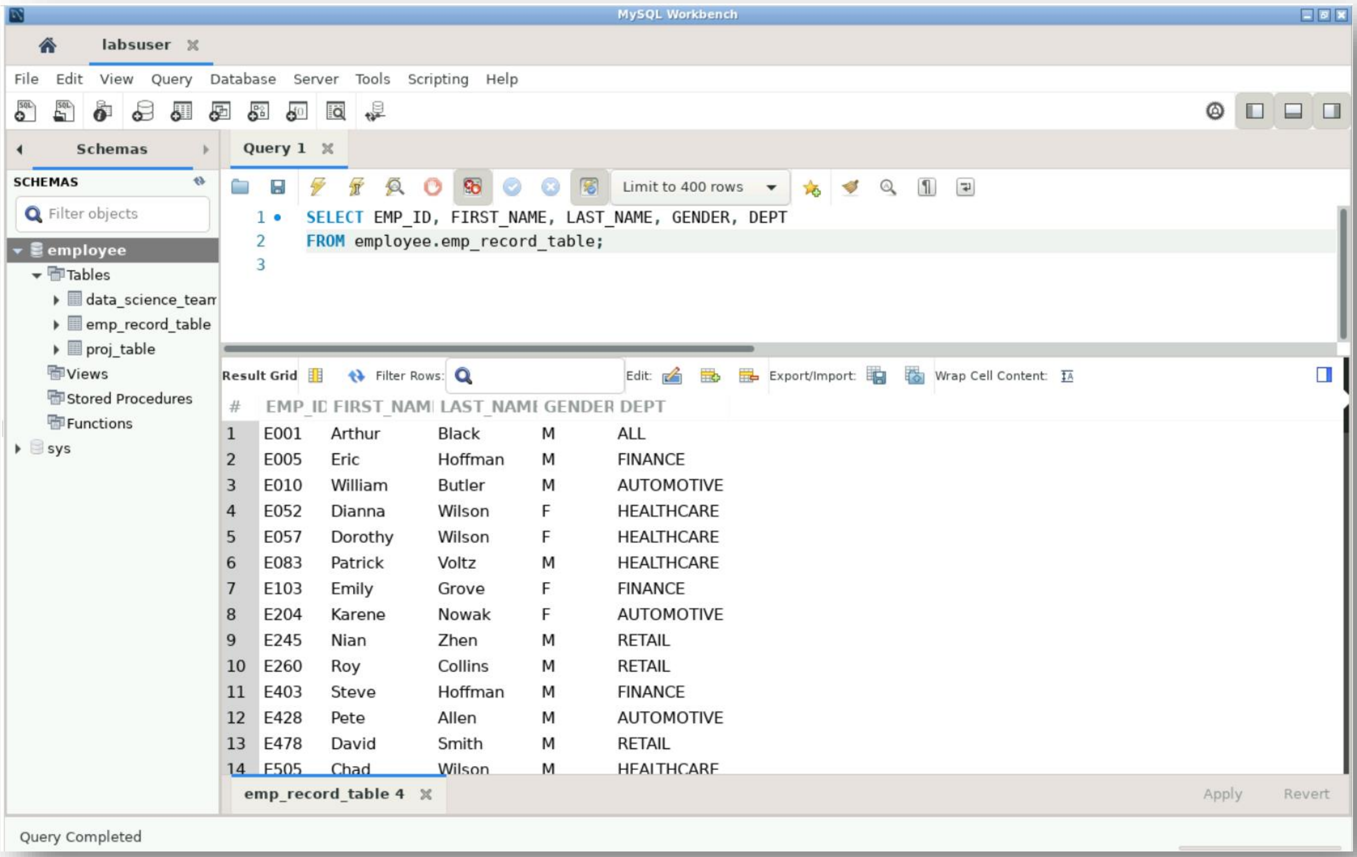


3. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

Query-

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

Output-



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

#	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
1	E001	Arthur	Black	M	ALL
2	E005	Eric	Hoffman	M	FINANCE
3	E010	William	Butler	M	AUTOMOTIVE
4	E052	Dianna	Wilson	F	HEALTHCARE
5	E057	Dorothy	Wilson	F	HEALTHCARE
6	E083	Patrick	Voltz	M	HEALTHCARE
7	E103	Emily	Grove	F	FINANCE
8	E204	Karene	Nowak	F	AUTOMOTIVE
9	E245	Nian	Zhen	M	RETAIL
10	E260	Roy	Collins	M	RETAIL
11	E403	Steve	Hoffman	M	FINANCE
12	E428	Pete	Allen	M	AUTOMOTIVE
13	E478	David	Smith	M	RETAIL
14	E505	Chad	Wilson	M	HEALTHCARE

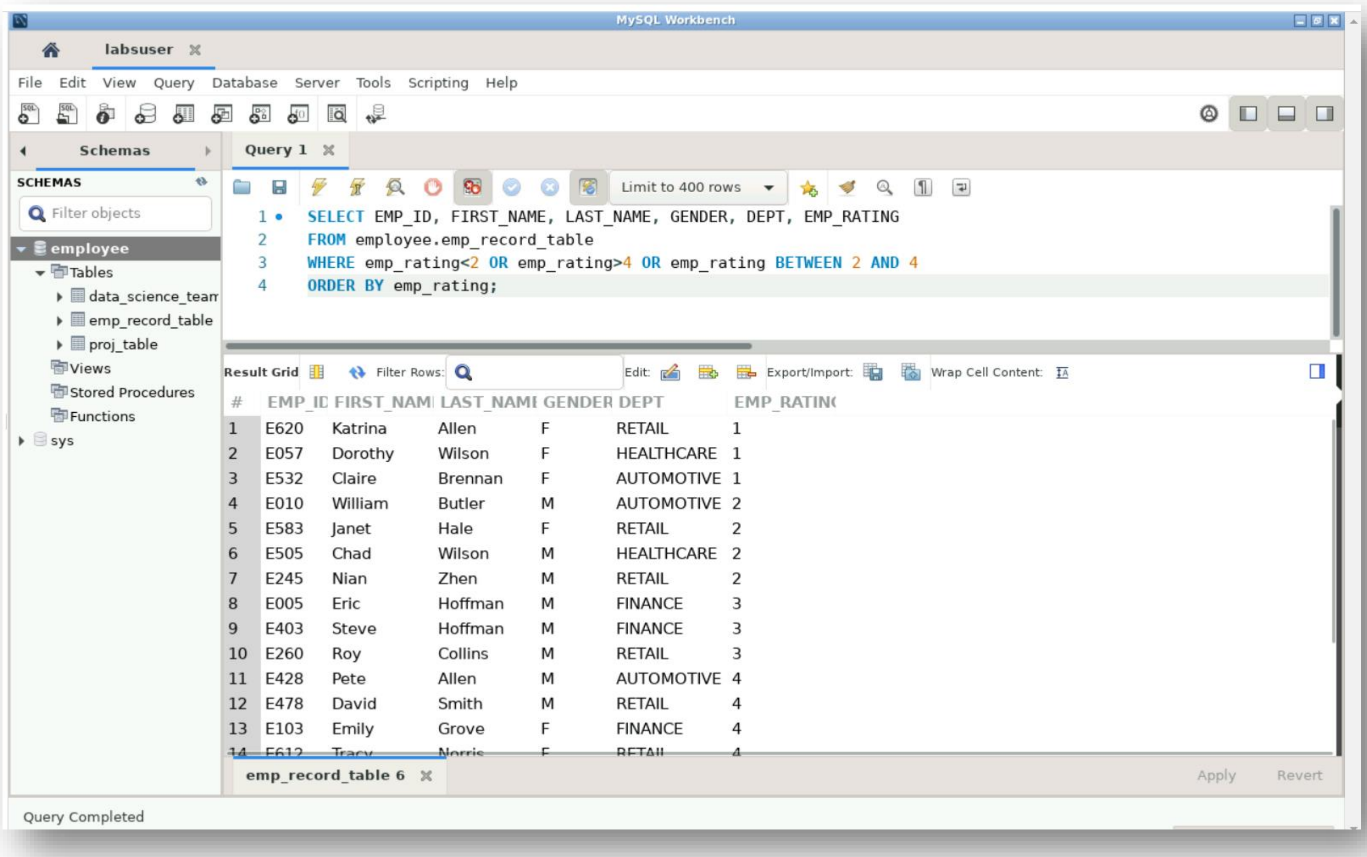
Query Completed

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-



```
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;
```

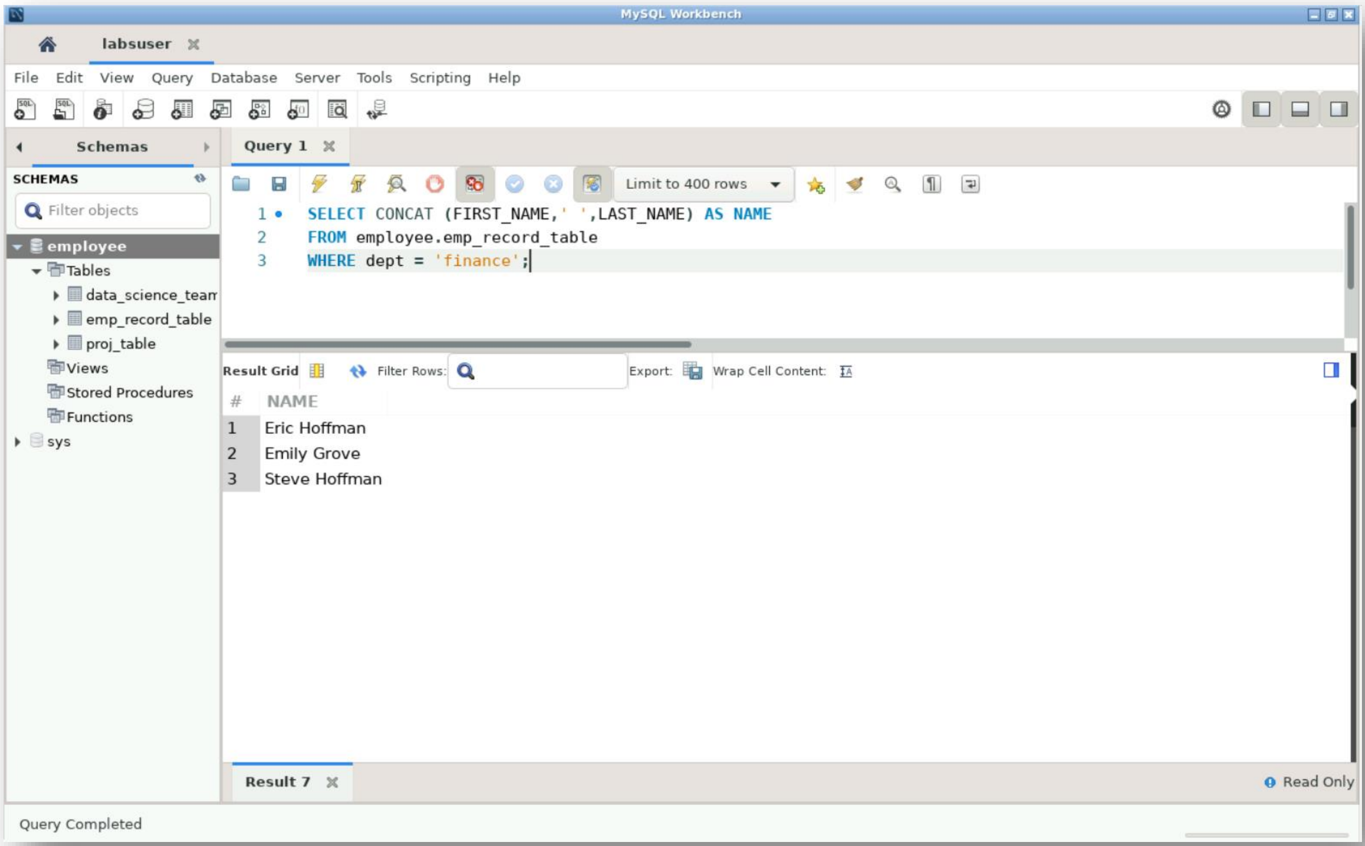
#	EMP_ID	FIRST_NAM	LAST_NAMI	GENDER	DEPT	EMP_RATIN
1	E620	Katrina	Allen	F	RETAIL	1
2	E057	Dorothy	Wilson	F	HEALTHCARE	1
3	E532	Claire	Brennan	F	AUTOMOTIVE	1
4	E010	William	Butler	M	AUTOMOTIVE	2
5	E583	Janet	Hale	F	RETAIL	2
6	E505	Chad	Wilson	M	HEALTHCARE	2
7	E245	Nian	Zhen	M	RETAIL	2
8	E005	Eric	Hoffman	M	FINANCE	3
9	E403	Steve	Hoffman	M	FINANCE	3
10	E260	Roy	Collins	M	RETAIL	3
11	E428	Pete	Allen	M	AUTOMOTIVE	4
12	E478	David	Smith	M	RETAIL	4
13	E103	Emily	Grove	F	FINANCE	4
14	E612	Tracy	Norris	F	RETAIL	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';

Output-



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

```
#   NAME
1   Eric Hoffman
2   Emily Grove
3   Steve Hoffman
```

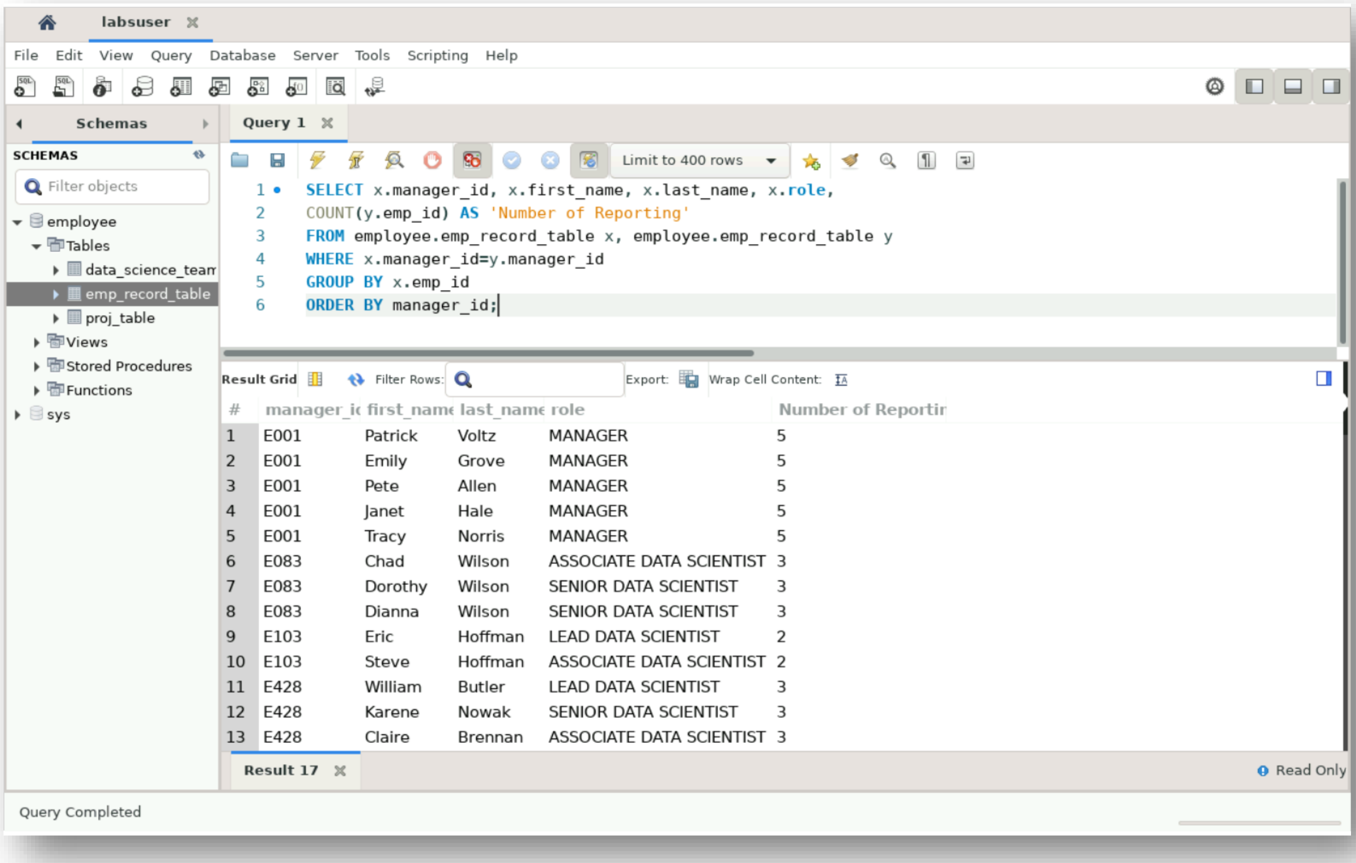
Query Completed

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-



```
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;
```

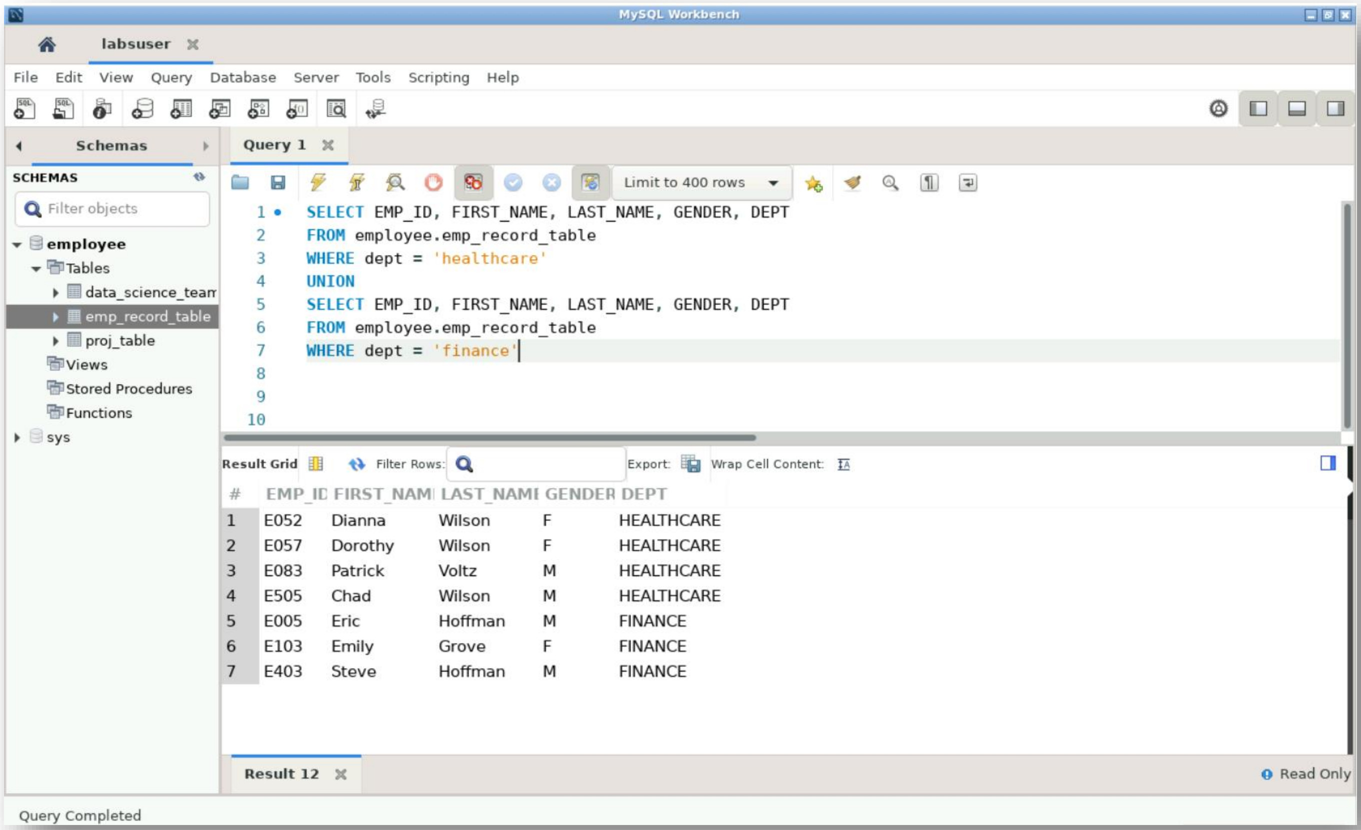
#	manager_ic	first_name	last_name	role	Number of Reportir
1	E001	Patrick	Voltz	MANAGER	5
2	E001	Emily	Grove	MANAGER	5
3	E001	Pete	Allen	MANAGER	5
4	E001	Janet	Hale	MANAGER	5
5	E001	Tracy	Norris	MANAGER	5
6	E083	Chad	Wilson	ASSOCIATE DATA SCIENTIST	3
7	E083	Dorothy	Wilson	SENIOR DATA SCIENTIST	3
8	E083	Dianna	Wilson	SENIOR DATA SCIENTIST	3
9	E103	Eric	Hoffman	LEAD DATA SCIENTIST	2
10	E103	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	2
11	E428	William	Butler	LEAD DATA SCIENTIST	3
12	E428	Karene	Nowak	SENIOR DATA SCIENTIST	3
13	E428	Claire	Brennan	ASSOCIATE DATA SCIENTIST	3

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-



```
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'
```

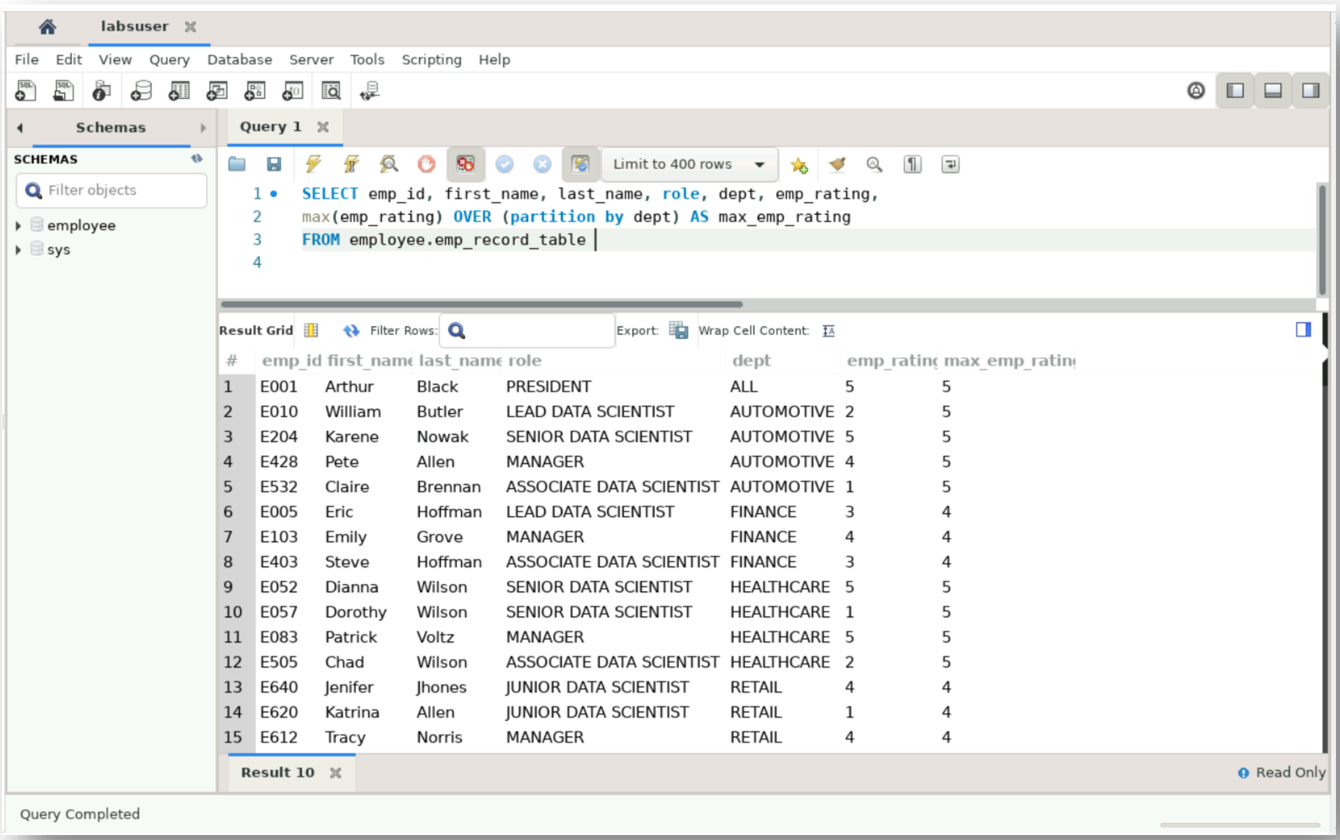
#	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
1	E052	Dianna	Wilson	F	HEALTHCARE
2	E057	Dorothy	Wilson	F	HEALTHCARE
3	E083	Patrick	Voltz	M	HEALTHCARE
4	E505	Chad	Wilson	M	HEALTHCARE
5	E005	Eric	Hoffman	M	FINANCE
6	E103	Emily	Grove	F	FINANCE
7	E403	Steve	Hoffman	M	FINANCE

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

Output-



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

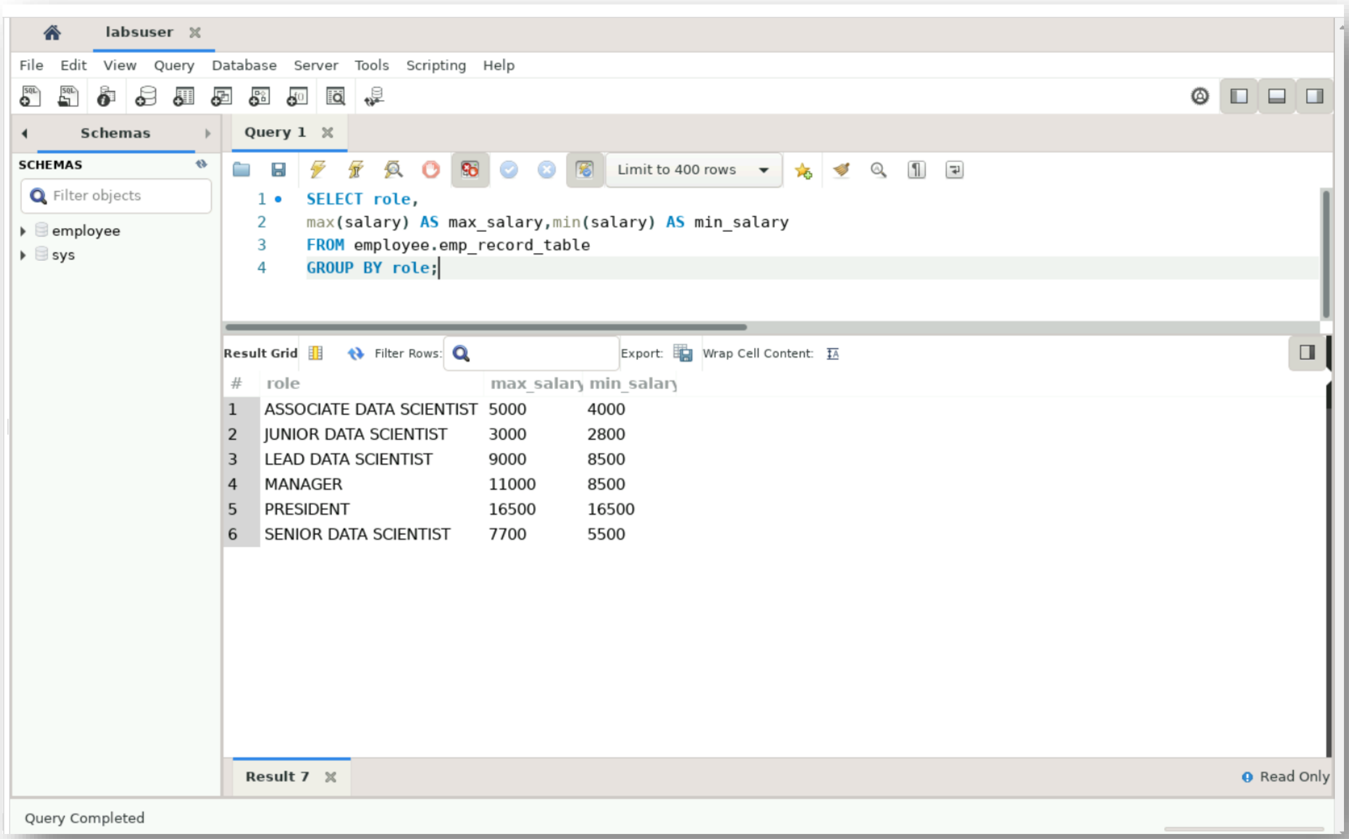
#	emp_id	first_name	last_name	role	dept	emp_rating	max_emp_rating
1	E001	Arthur	Black	PRESIDENT	ALL	5	5
2	E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
3	E204	Karene	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
4	E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
5	E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
6	E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4
7	E103	Emily	Grove	MANAGER	FINANCE	4	4
8	E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
9	E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	5	5
10	E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	1	5
11	E083	Patrick	Voltz	MANAGER	HEALTHCARE	5	5
12	E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
13	E640	Jenifer	Jhones	JUNIOR DATA SCIENTIST	RETAIL	4	4
14	E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	1	4
15	E612	Tracy	Norris	MANAGER	RETAIL	4	4

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-



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

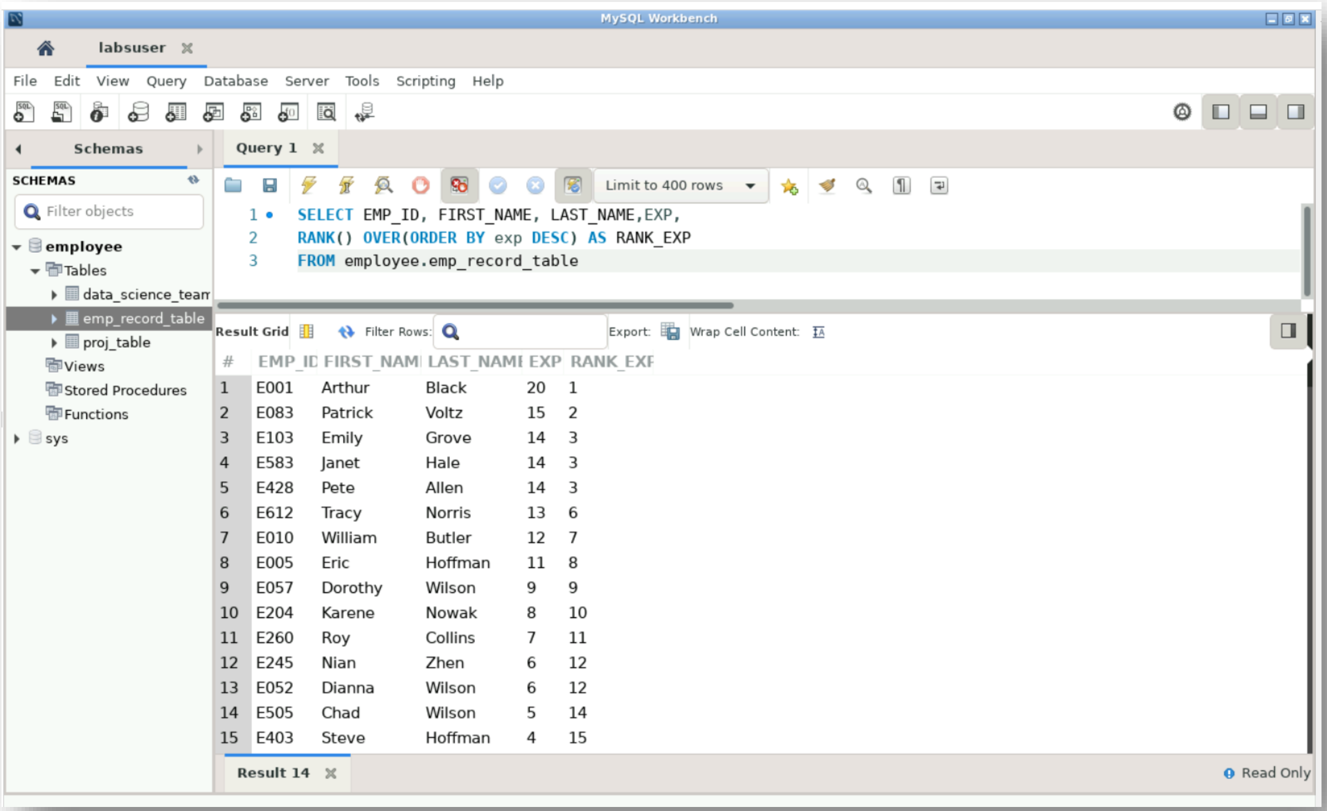
```
#  role                     max_salary min_salary
1  ASSOCIATE DATA SCIENTIST 5000       4000
2  JUNIOR DATA SCIENTIST    3000       2800
3  LEAD DATA SCIENTIST      9000       8500
4  MANAGER                  11000      8500
5  PRESIDENT                16500      16500
6  SENIOR DATA SCIENTIST    7700       5500
```

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-



```
SELECT EMP_ID, FIRST_NAME, LAST_NAME,EXP,
RANK() OVER(ORDER BY exp DESC) AS RANK_EXP
FROM employee.emp_record_table
```

#	EMP_ID	FIRST_NAME	LAST_NAME	EXP	RANK_EXP
1	E001	Arthur	Black	20	1
2	E083	Patrick	Voltz	15	2
3	E103	Emily	Grove	14	3
4	E583	Janet	Hale	14	3
5	E428	Pete	Allen	14	3
6	E612	Tracy	Norris	13	6
7	E010	William	Butler	12	7
8	E005	Eric	Hoffman	11	8
9	E057	Dorothy	Wilson	9	9
10	E204	Karene	Nowak	8	10
11	E260	Roy	Collins	7	11
12	E245	Nian	Zhen	6	12
13	E052	Dianna	Wilson	6	12
14	E505	Chad	Wilson	5	14
15	E403	Steve	Hoffman	4	15

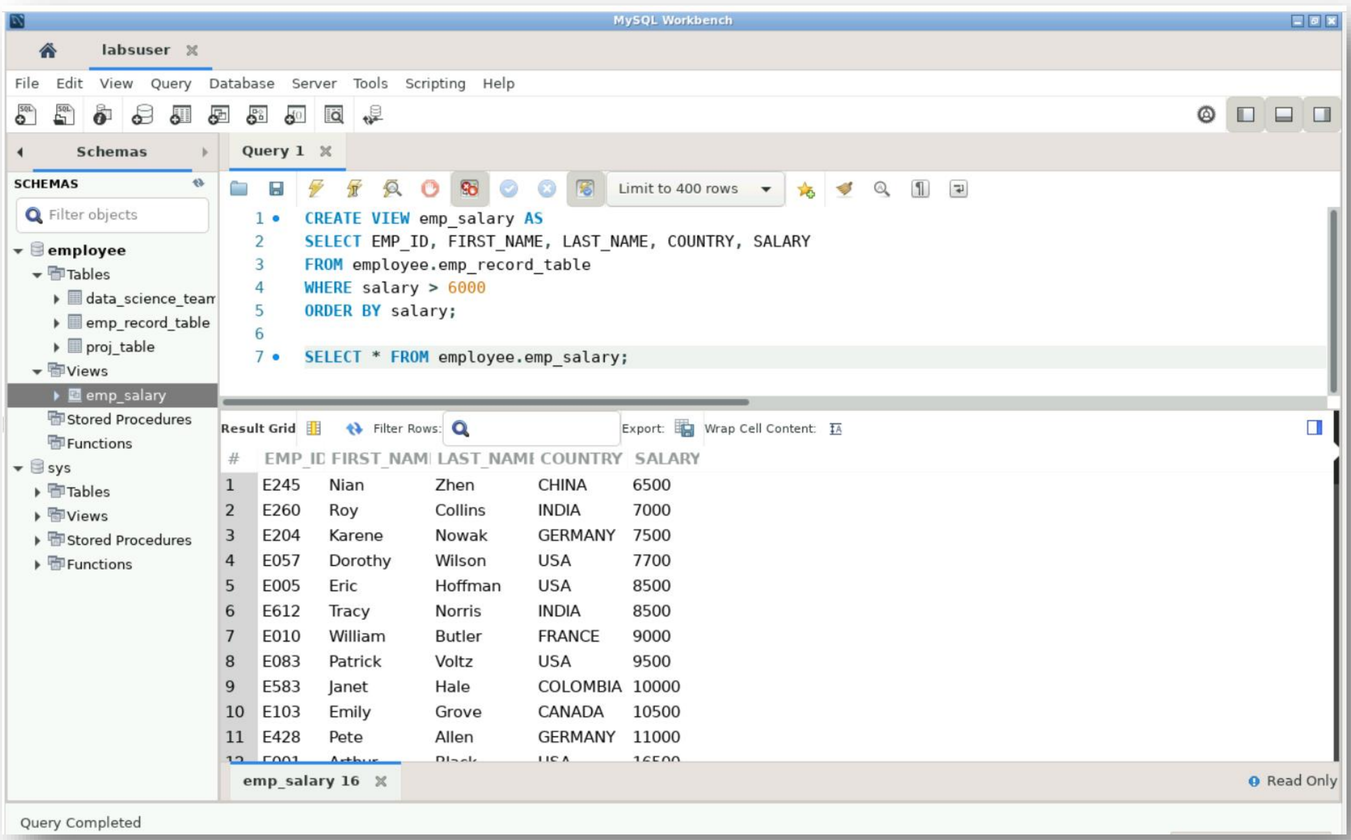
11. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.

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-



```
CREATE VIEW emp_salary AS
SELECT EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY
FROM employee.emp_record_table
WHERE salary > 6000
ORDER BY salary;

SELECT * FROM employee.emp_salary;
```

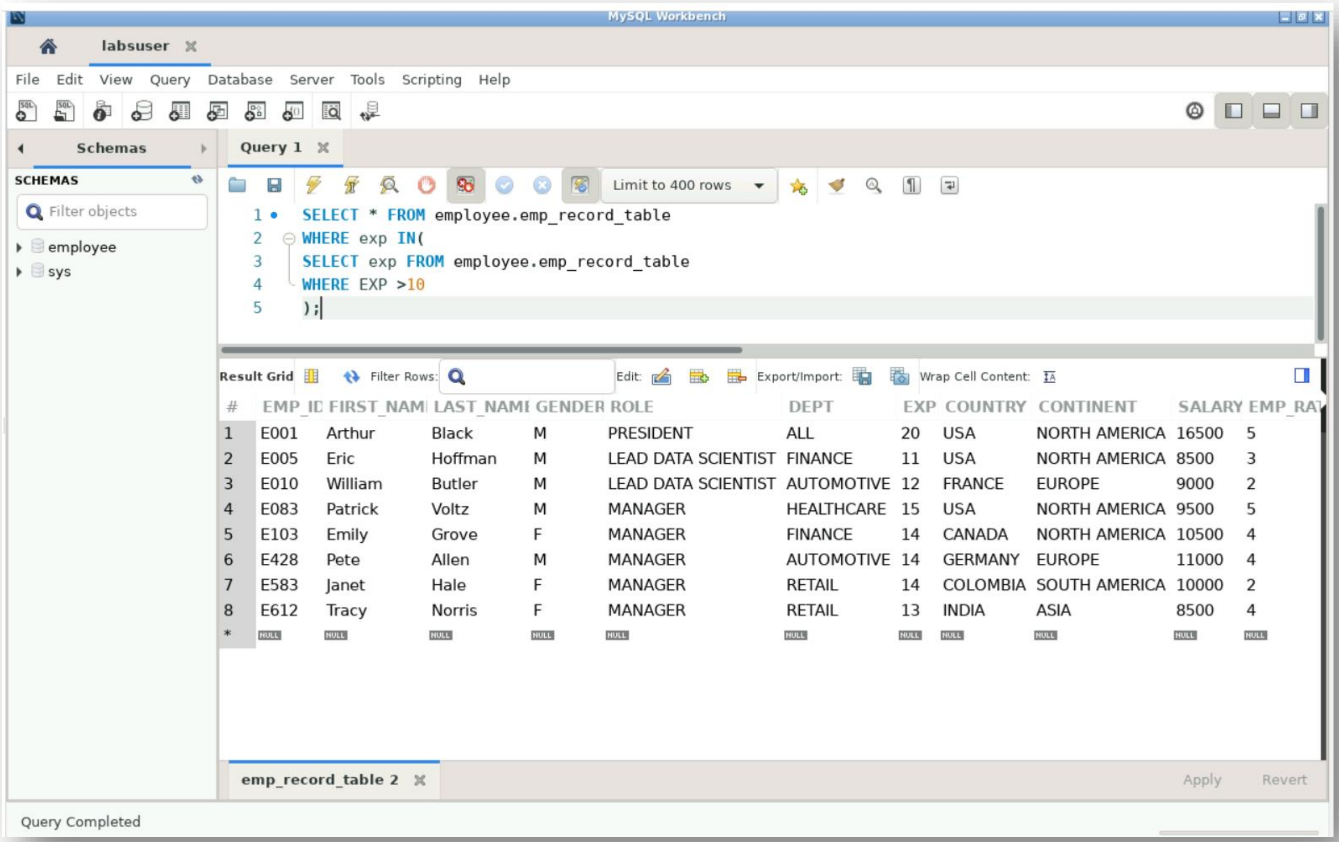
#	EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
1	E245	Nian	Zhen	CHINA	6500
2	E260	Roy	Collins	INDIA	7000
3	E204	Karene	Nowak	GERMANY	7500
4	E057	Dorothy	Wilson	USA	7700
5	E005	Eric	Hoffman	USA	8500
6	E612	Tracy	Norris	INDIA	8500
7	E010	William	Butler	FRANCE	9000
8	E083	Patrick	Voltz	USA	9500
9	E583	Janet	Hale	COLOMBIA	10000
10	E103	Emily	Grove	CANADA	10500
11	E428	Pete	Allen	GERMANY	11000

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-



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

#	EMP_ID	FIRST_NAM	LAST_NAMI	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RA
1	E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500	5
2	E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3
3	E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2
4	E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5
5	E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4
6	E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4
7	E583	Janet	Hale	F	MANAGER	RETAIL	14	COLOMBIA	SOUTH AMERICA	10000	2
8	E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

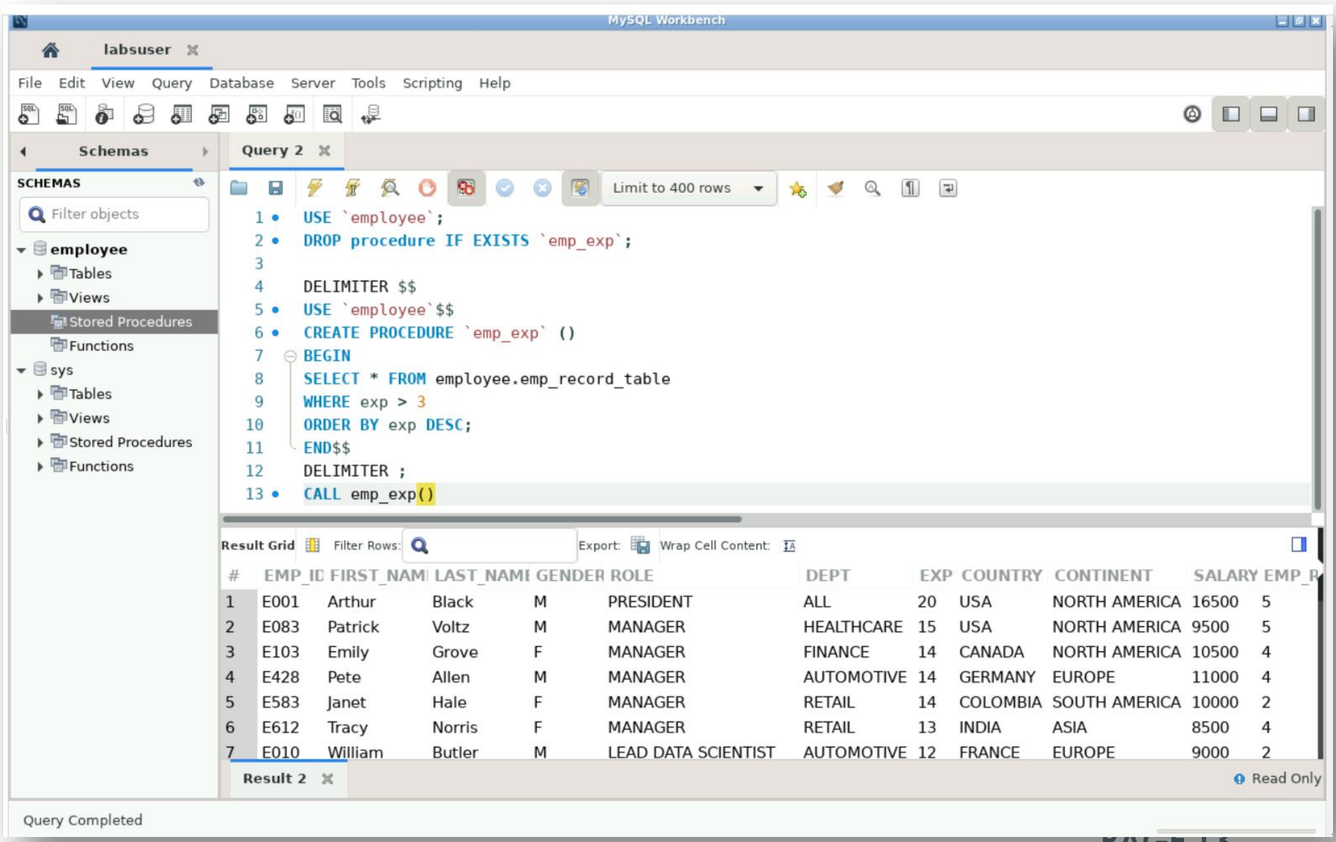
Query Completed

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-



```
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()
```

#	EMP_ID	FIRST_NAM	LAST_NAM	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_R
1	E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500	5
2	E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5
3	E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4
4	E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4
5	E583	Janet	Hale	F	MANAGER	RETAIL	14	COLOMBIA	SOUTH AMERICA	10000	2
6	E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4
7	E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2

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:

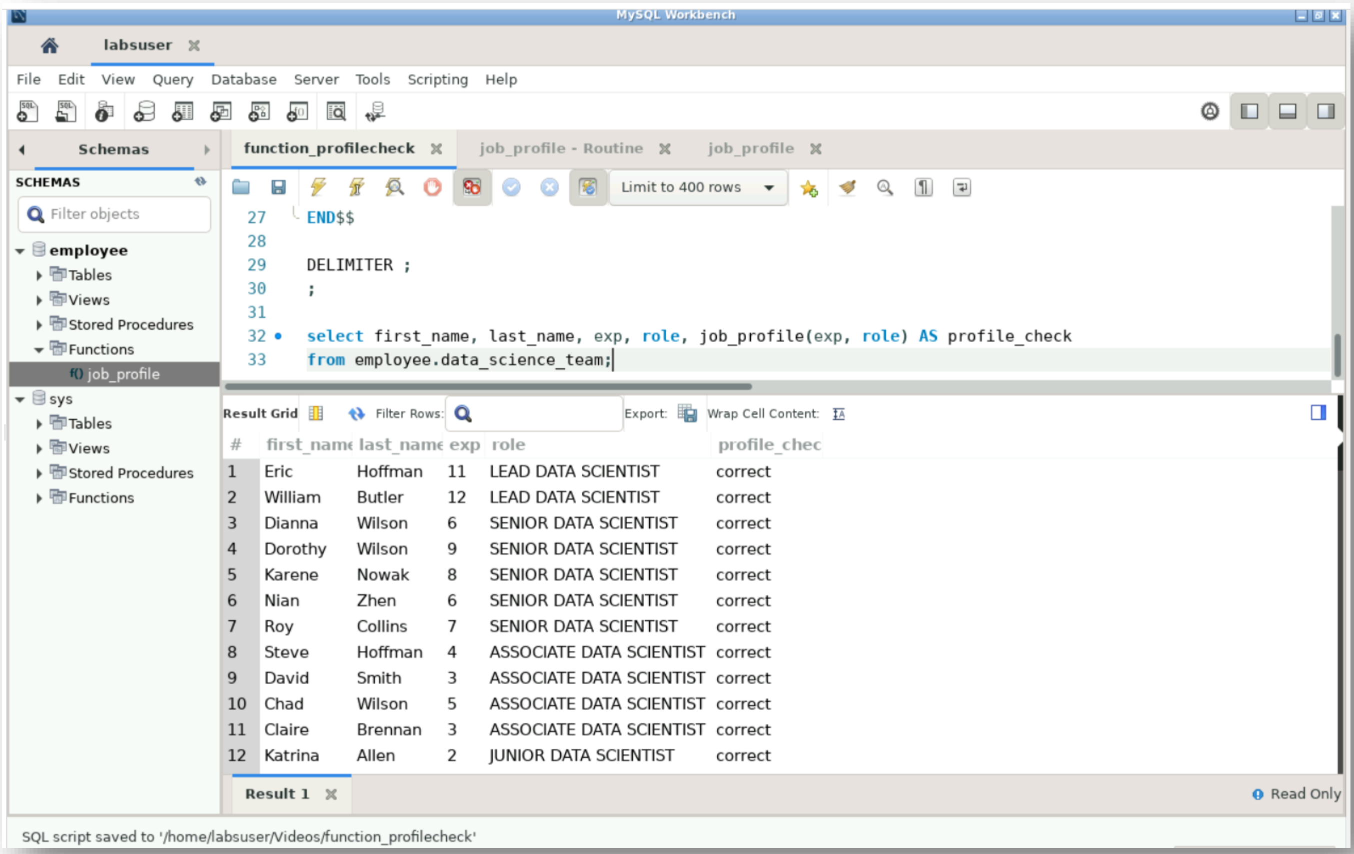
For an employee with experience less than or equal to 2 years assign 'JUNIOR DATA SCIENTIST',
For an employee with the experience of 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST',
For an employee with the experience of 5 to 10 years assign 'SENIOR DATA SCIENTIST',
For an employee with the experience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



```
END$$

DELIMITER ;
;

select first_name, last_name, exp, role, job_profile(exp, role) AS profile_check
from employee.data_science_team;
```

#	first_name	last_name	exp	role	profile_chec
1	Eric	Hoffman	11	LEAD DATA SCIENTIST	correct
2	William	Butler	12	LEAD DATA SCIENTIST	correct
3	Dianna	Wilson	6	SENIOR DATA SCIENTIST	correct
4	Dorothy	Wilson	9	SENIOR DATA SCIENTIST	correct
5	Karene	Nowak	8	SENIOR DATA SCIENTIST	correct
6	Nian	Zhen	6	SENIOR DATA SCIENTIST	correct
7	Roy	Collins	7	SENIOR DATA SCIENTIST	correct
8	Steve	Hoffman	4	ASSOCIATE DATA SCIENTIST	correct
9	David	Smith	3	ASSOCIATE DATA SCIENTIST	correct
10	Chad	Wilson	5	ASSOCIATE DATA SCIENTIST	correct
11	Claire	Brennan	3	ASSOCIATE DATA SCIENTIST	correct
12	Katrina	Allen	2	JUNIOR DATA SCIENTIST	correct

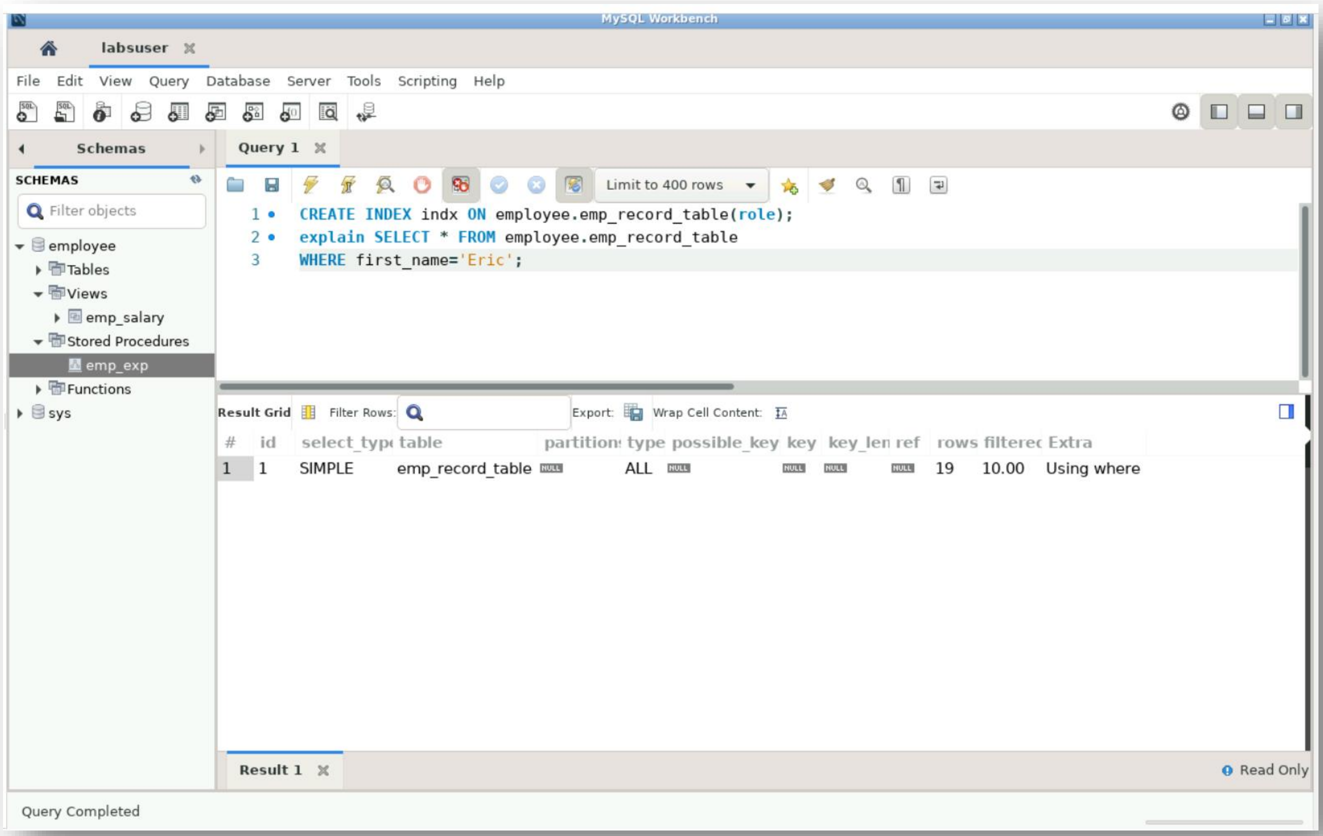
SQL script saved to '/home/labsuser/Videos/function_profilecheck'

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-



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

```
#  id  select_type  table             partitions  type  possible_key  key   key_len  ref   rows  filtered  Extra
1  1   SIMPLE       emp_record_table  NULL        ALL   NULL          NULL  NULL     NULL  19    10.00     Using where
```

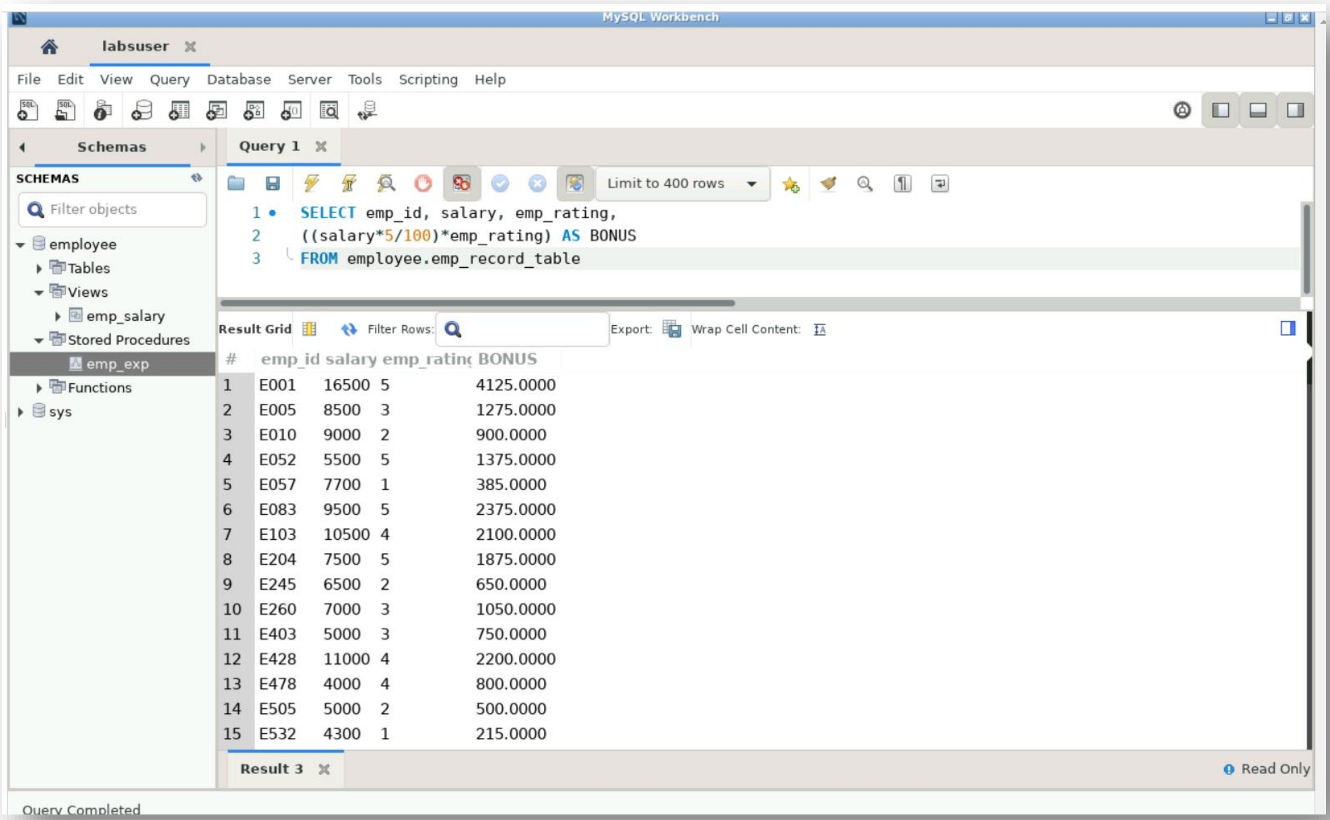
Query Completed

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-



```
SELECT emp_id, salary, emp_rating,
((salary*5/100)*emp_rating) AS BONUS
FROM employee.emp_record_table
```

#	emp_id	salary	emp_rating	BONUS
1	E001	16500	5	4125.0000
2	E005	8500	3	1275.0000
3	E010	9000	2	900.0000
4	E052	5500	5	1375.0000
5	E057	7700	1	385.0000
6	E083	9500	5	2375.0000
7	E103	10500	4	2100.0000
8	E204	7500	5	1875.0000
9	E245	6500	2	650.0000
10	E260	7000	3	1050.0000
11	E403	5000	3	750.0000
12	E428	11000	4	2200.0000
13	E478	4000	4	800.0000
14	E505	5000	2	500.0000
15	E532	4300	1	215.0000

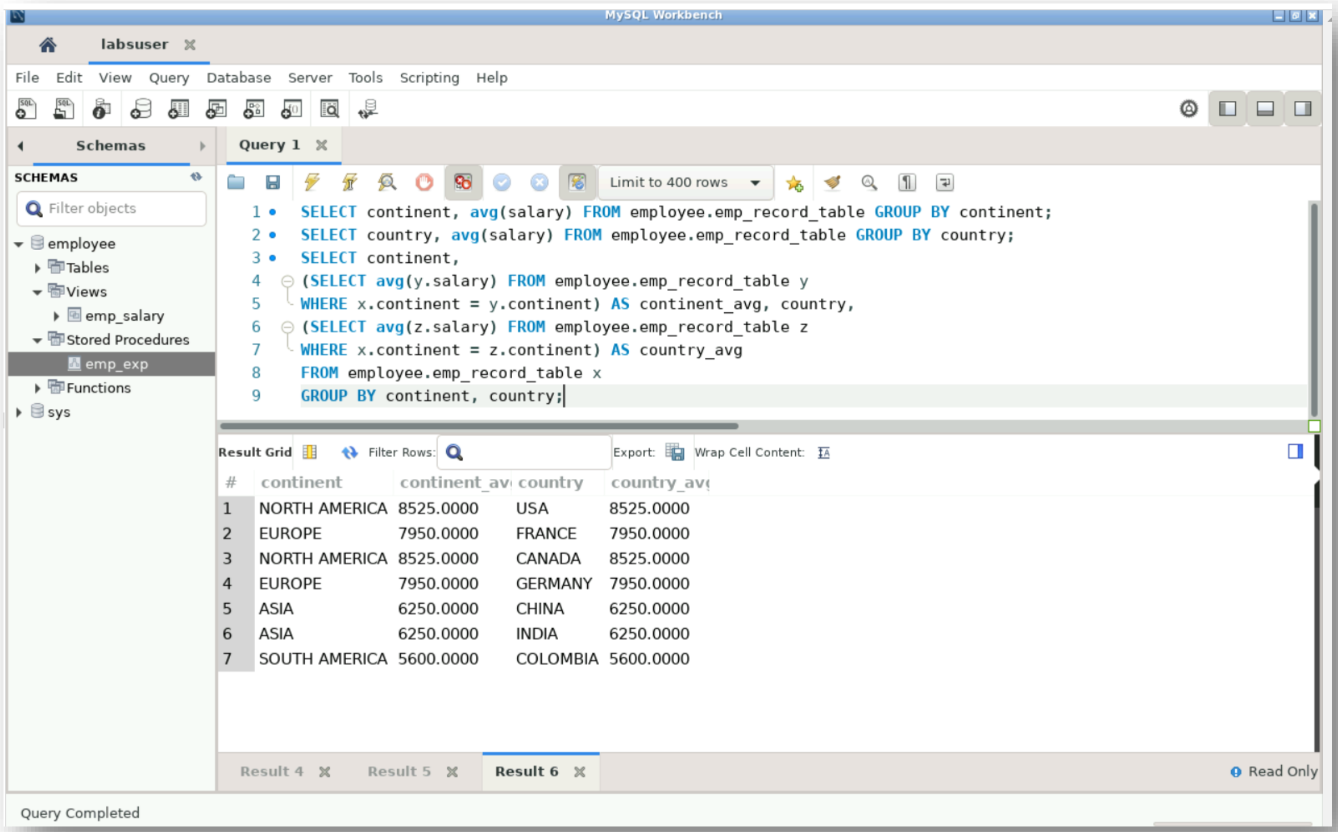
Query Completed

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-



```
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;
```

#	continent	continent_av	country	country_av
1	NORTH AMERICA	8525.0000	USA	8525.0000
2	EUROPE	7950.0000	FRANCE	7950.0000
3	NORTH AMERICA	8525.0000	CANADA	8525.0000
4	EUROPE	7950.0000	GERMANY	7950.0000
5	ASIA	6250.0000	CHINA	6250.0000
6	ASIA	6250.0000	INDIA	6250.0000
7	SOUTH AMERICA	5600.0000	COLOMBIA	5600.0000

Query Completed