**Author:** Divya Pardeshi

**Date:** 22-11-2023

# ScienceQtech Employee Performance Mapping- SQL Project

**Q1) 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.**

**Ans.** #Create the employee database if it doesn't exist

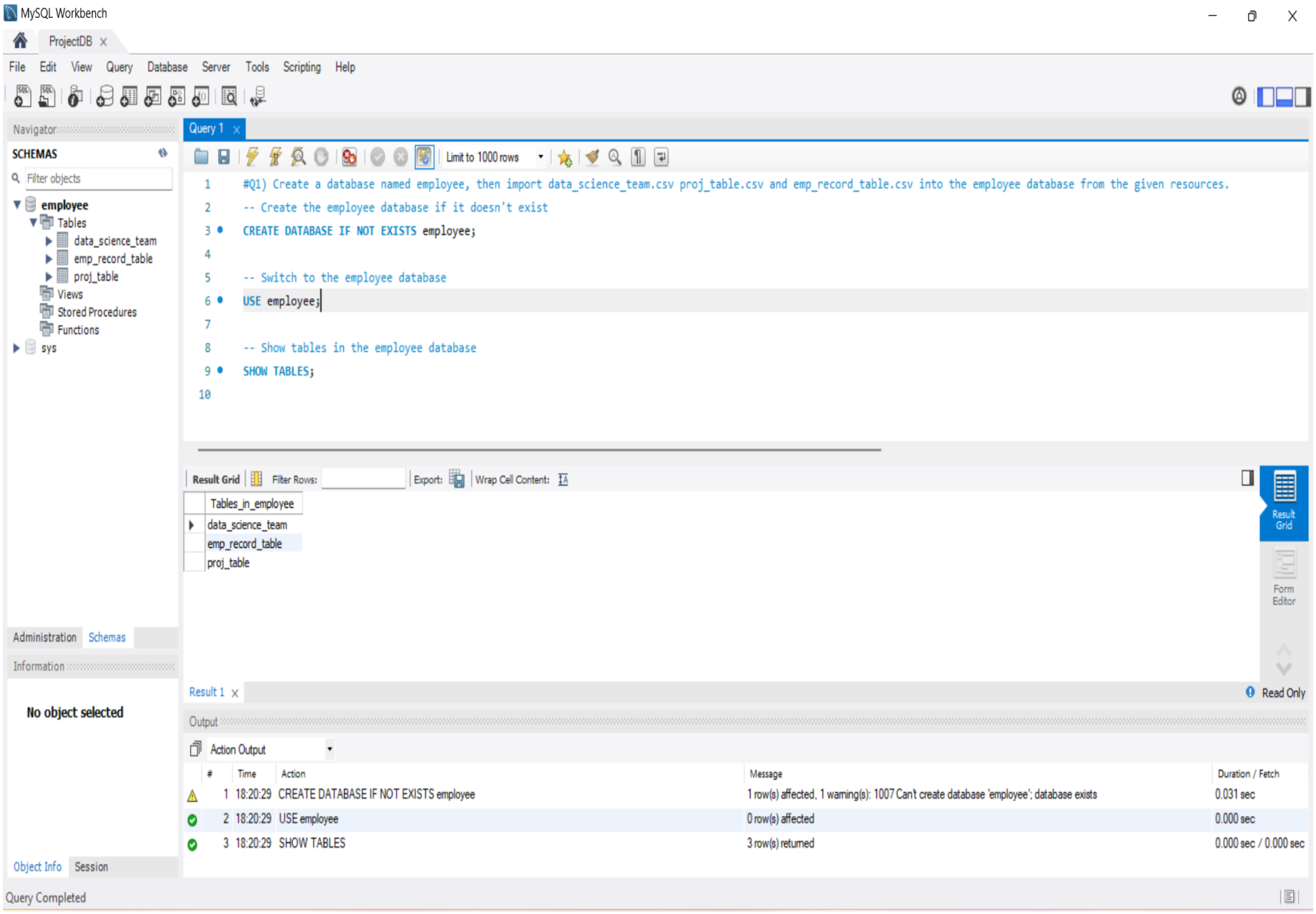
CREATE DATABASE IF NOT EXISTS employee;

#Switch to the employee database

USE employee;

#Show tables in the employee database

SHOW TABLES;



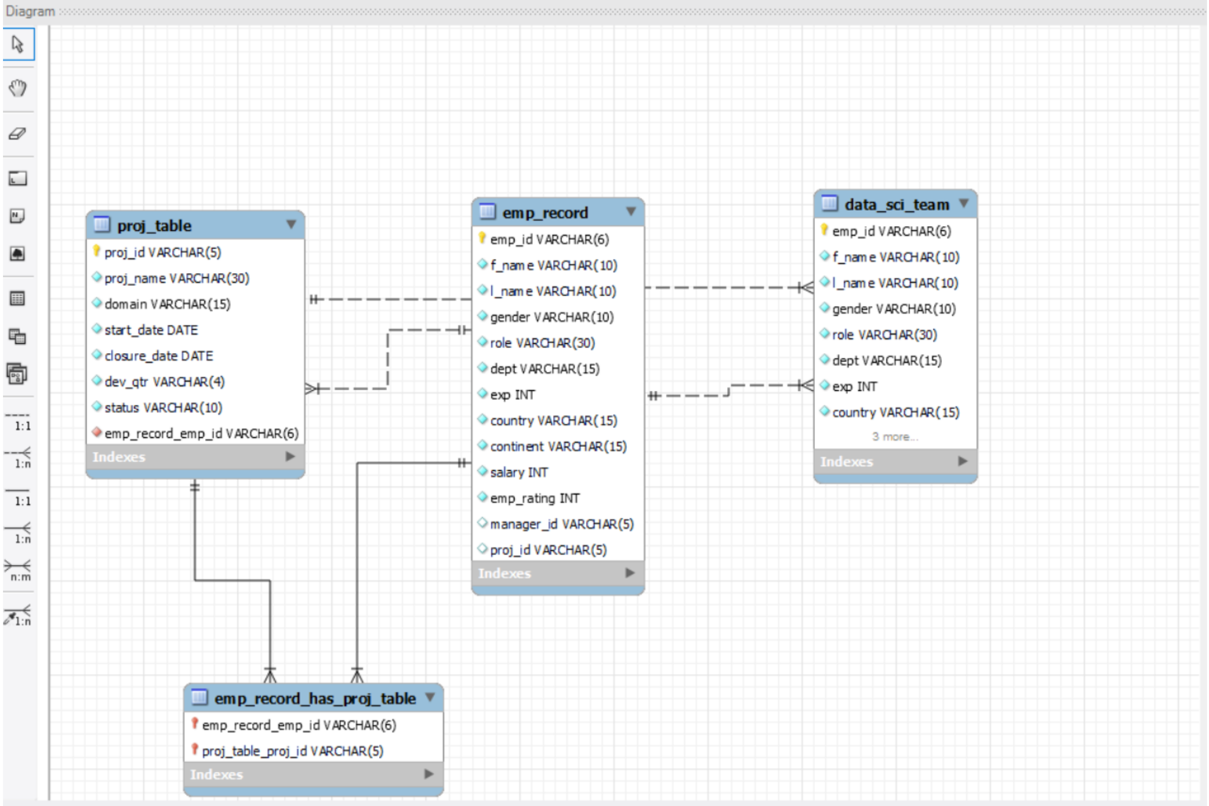
```
#Q1) Create a database named employee, then import data_science_team.csv proj_table.csv and emp_record_table.csv into the employee database from the given resources.
-- Create the employee database if it doesn't exist
CREATE DATABASE IF NOT EXISTS employee;

-- Switch to the employee database
USE employee;

-- Show tables in the employee database
SHOW TABLES;
```

| Tables_in_employee |
|---|
| data_science_team |
| emp_record_table |
| proj_table |

| | # | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|---|
| ⚠ | 1 | 18:20:29 | CREATE DATABASE IF NOT EXISTS employee | 1 row(s) affected, 1 warning(s): 1007 Can't create database 'employee'; database exists | 0.031 sec |
| ✓ | 2 | 18:20:29 | USE employee | 0 row(s) affected | 0.000 sec |
| ✓ | 3 | 18:20:29 | SHOW TABLES | 3 row(s) returned | 0.000 sec / 0.000 sec |

**Q2) Create an ER diagram for the given employee database.**

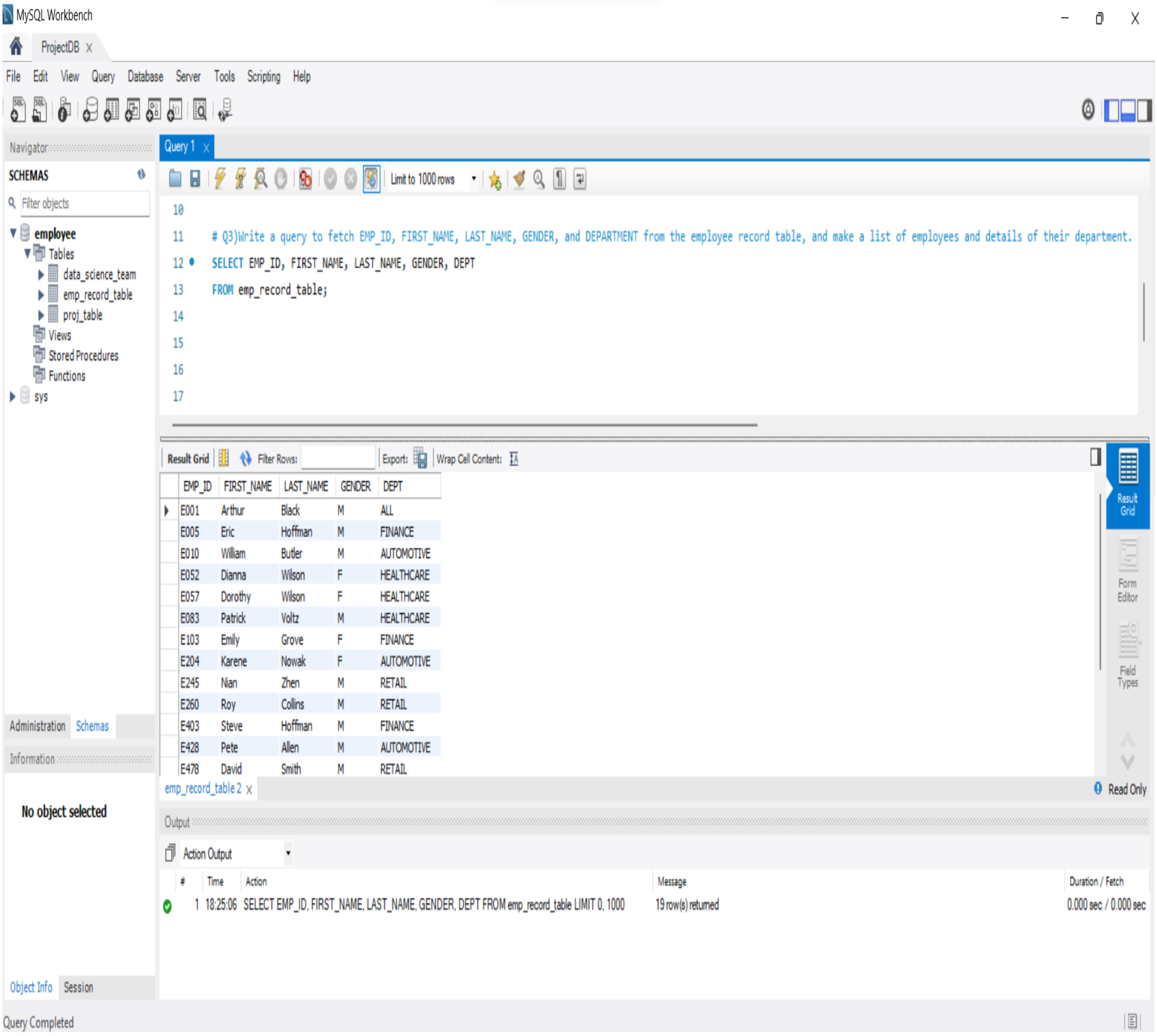
**Ans.**



**Q3) 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.**

**Ans.** SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT

FROM emp_record_table;



```
# Q3)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.
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM emp_record_table;
```

| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | DEPT |
|---|---|---|---|---|
| E001 | Arthur | Black | M | ALL |
| E005 | Eric | Hoffman | M | FINANCE |
| E010 | William | Butler | M | AUTOMOTIVE |
| E052 | Dianna | Wilson | F | HEALTHCARE |
| E057 | Dorothy | Wilson | F | HEALTHCARE |
| E083 | Patrick | Voltz | M | HEALTHCARE |
| E103 | Emily | Grove | F | FINANCE |
| E204 | Karene | Nowak | F | AUTOMOTIVE |
| E245 | Nian | Zhen | M | RETAIL |
| E260 | Roy | Collins | M | RETAIL |
| E403 | Steve | Hoffman | M | FINANCE |
| E428 | Pete | Allen | M | AUTOMOTIVE |
| E478 | David | Smith | M | RETAIL |

| # | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| 1 | 18:25:06 | SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table LIMIT 0, 1000 | 19 row(s) returned | 0.000 sec / 0.000 sec |

**Q4) Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:**

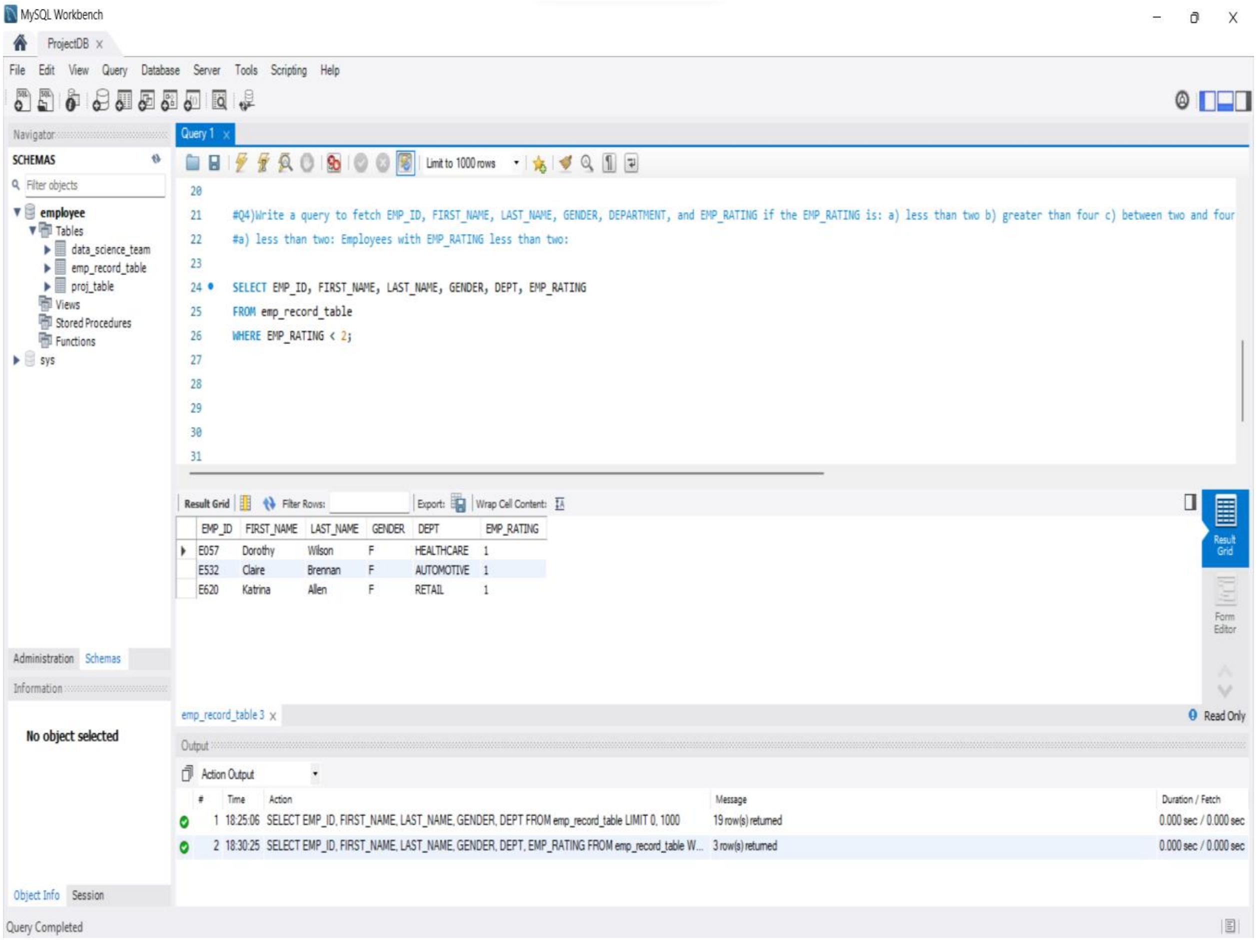
- **less than two**
- **greater than four**
- **between two and four**

**Ans. a) less than two:** Employees with EMP_RATING less than two:

SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING

FROM emp_record_table

WHERE EMP_RATING < 2;



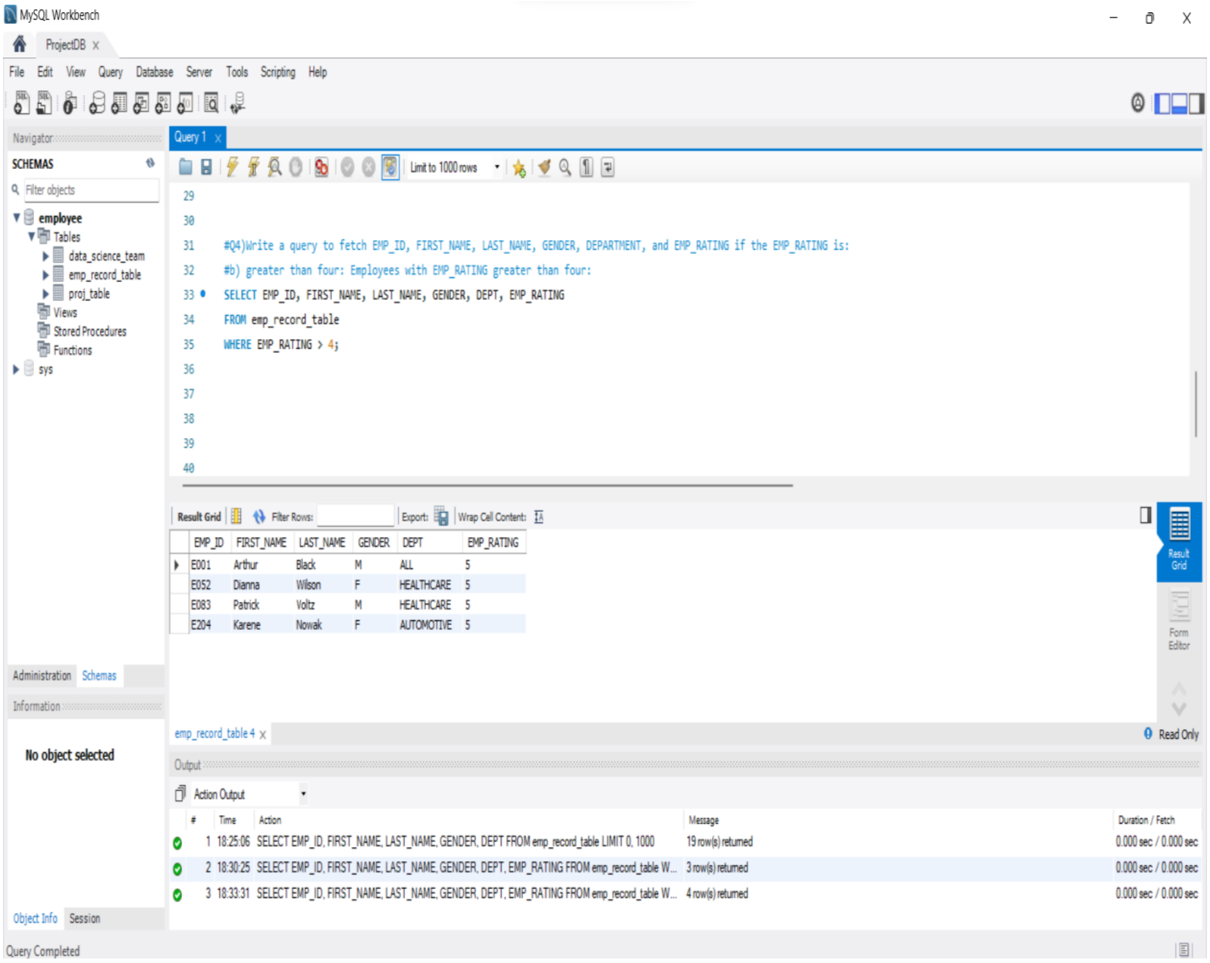
```
#Q4)Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is: a) less than two b) greater than four c) between two and four
#a) less than two: Employees with EMP_RATING less than two:

SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING < 2;
```

| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | DEPT | EMP_RATING |
|---|---|---|---|---|---|
| E057 | Dorothy | Wilson | F | HEALTHCARE | 1 |
| E532 | Claire | Brennan | F | AUTOMOTIVE | 1 |
| E620 | Katrina | Allen | F | RETAIL | 1 |

```
#  Time      Action                                                                                  Message             Duration / Fetch
1  18:25:06  SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table LIMIT 0, 1000  19 row(s) returned  0.000 sec / 0.000 sec
2  18:30:25  SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING FROM emp_record_table W...  3 row(s) returned  0.000 sec / 0.000 sec
```

**b) greater than four:** Employees with EMP_RATING greater than four:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING

FROM emp_record_table

WHERE EMP_RATING > 4;
```



```
#Q4)Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:
#b) greater than four: Employees with EMP_RATING greater than four:
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING > 4;
```

| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | DEPT | EMP_RATING |
|---|---|---|---|---|---|
| E001 | Arthur | Black | M | ALL | 5 |
| E052 | Dianna | Wilson | F | HEALTHCARE | 5 |
| E083 | Patrick | Voltz | M | HEALTHCARE | 5 |
| E204 | Karene | Nowak | F | AUTOMOTIVE | 5 |

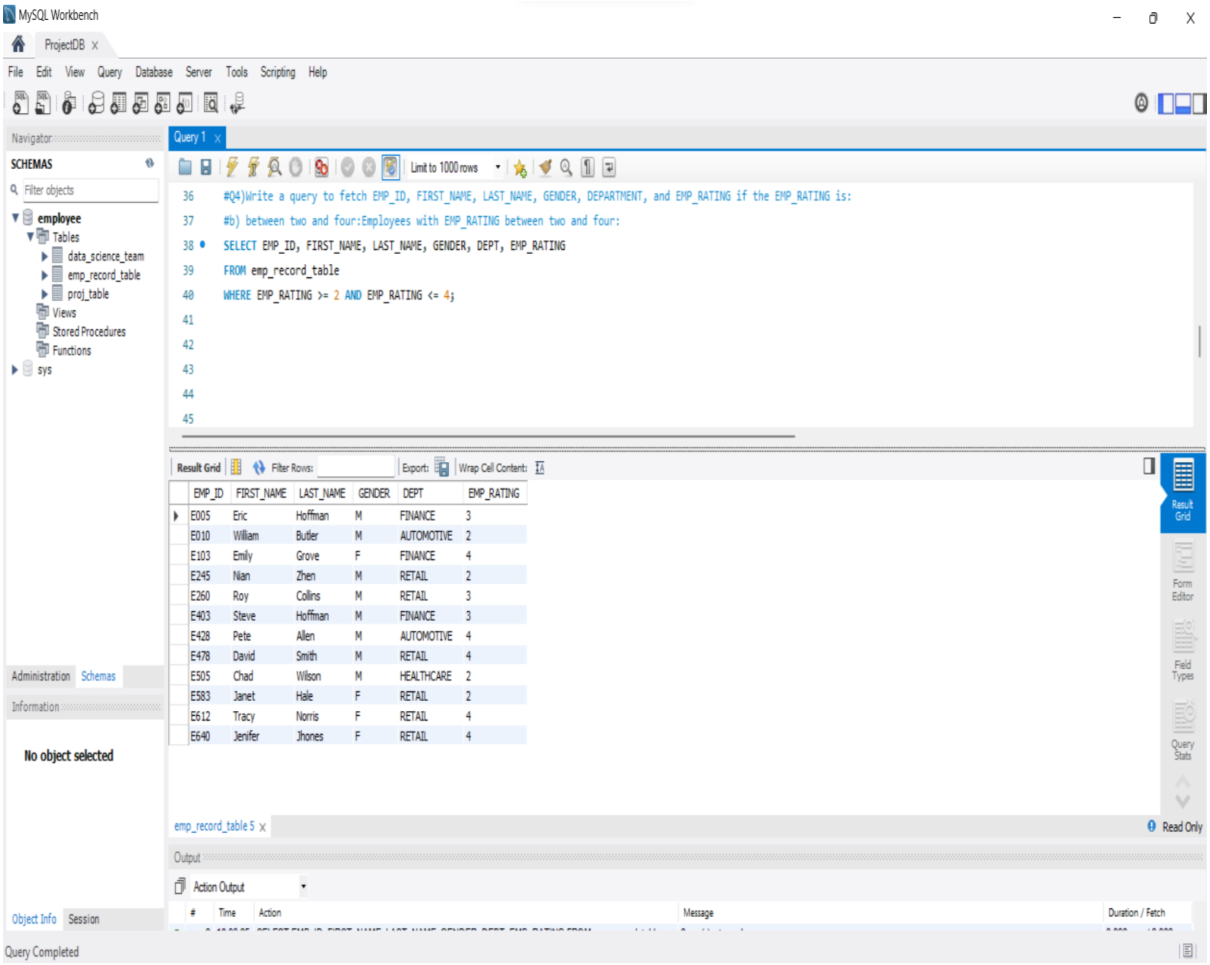
```
#  Time      Action                                                                         Message             Duration / Fetch
1  18:25:06  SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table LIMIT 0, 1000   19 row(s) returned   0.000 sec / 0.000 sec
2  18:30:25  SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING FROM emp_record_table W...   3 row(s) returned   0.000 sec / 0.000 sec
3  18:33:31  SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING FROM emp_record_table W...   4 row(s) returned   0.000 sec / 0.000 sec
```

**c) between two and four:** Employees with EMP_RATING between two and four:

SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING

FROM emp_record_table

WHERE EMP_RATING >= 2 AND EMP_RATING <= 4;



```
#Q4)Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:
#b) between two and four:Employees with EMP_RATING between two and four:
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING >= 2 AND EMP_RATING <= 4;
```

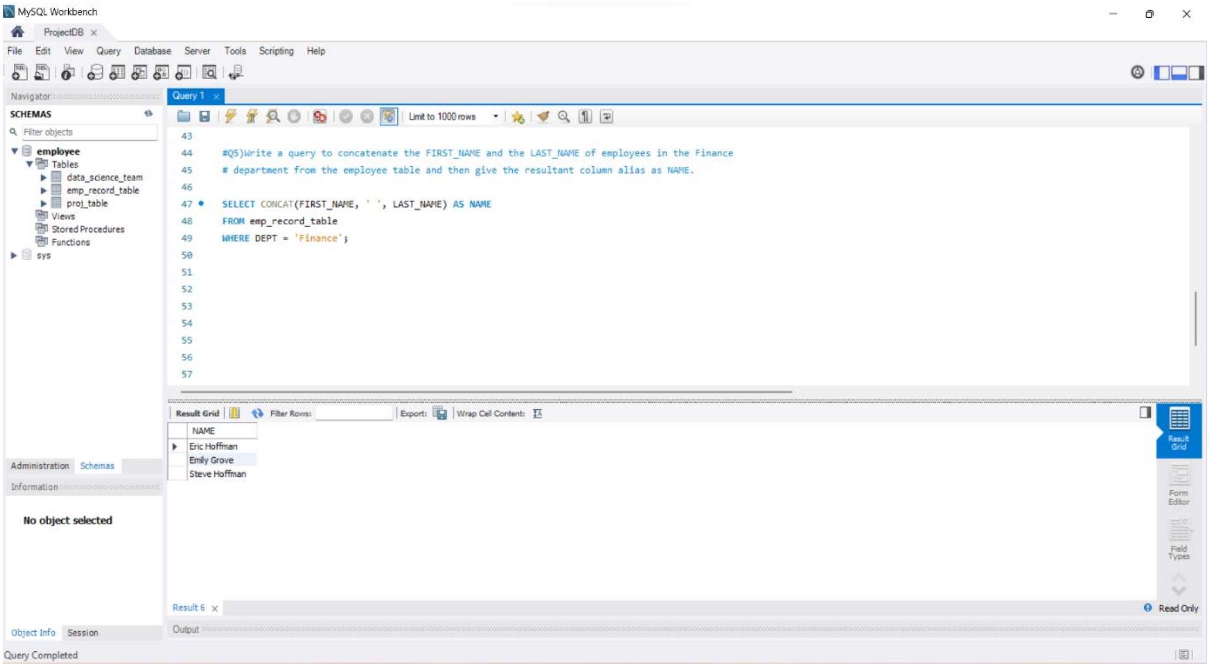
| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | DEPT | EMP_RATING |
|---|---|---|---|---|---|
| E005 | Eric | Hoffman | M | FINANCE | 3 |
| E010 | William | Butler | M | AUTOMOTIVE | 2 |
| E103 | Emily | Grove | F | FINANCE | 4 |
| E245 | Nian | Zhen | M | RETAIL | 2 |
| E260 | Roy | Collins | M | RETAIL | 3 |
| E403 | Steve | Hoffman | M | FINANCE | 3 |
| E428 | Pete | Allen | M | AUTOMOTIVE | 4 |
| E478 | David | Smith | M | RETAIL | 4 |
| E505 | Chad | Wilson | M | HEALTHCARE | 2 |
| E583 | Janet | Hale | F | RETAIL | 2 |
| E612 | Tracy | Norris | F | RETAIL | 4 |
| E640 | Jenifer | Jhones | F | RETAIL | 4 |

**Q5) 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.**

**Ans.** SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS NAME

FROM emp_record_table

WHERE DEPT = 'Finance';



```
#Q5)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.

SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS NAME
FROM emp_record_table
WHERE DEPT = 'Finance';
```

| NAME |
|---|
| Eric Hoffman |
| Emily Grove |
| Steve Hoffman |

**Q6) Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).**

**Ans.** SELECT E.EMP_ID, E.FIRST_NAME, E.LAST_NAME, E.ROLE, E.EXP,

(SELECT COUNT(*)

FROM emp_record_table R

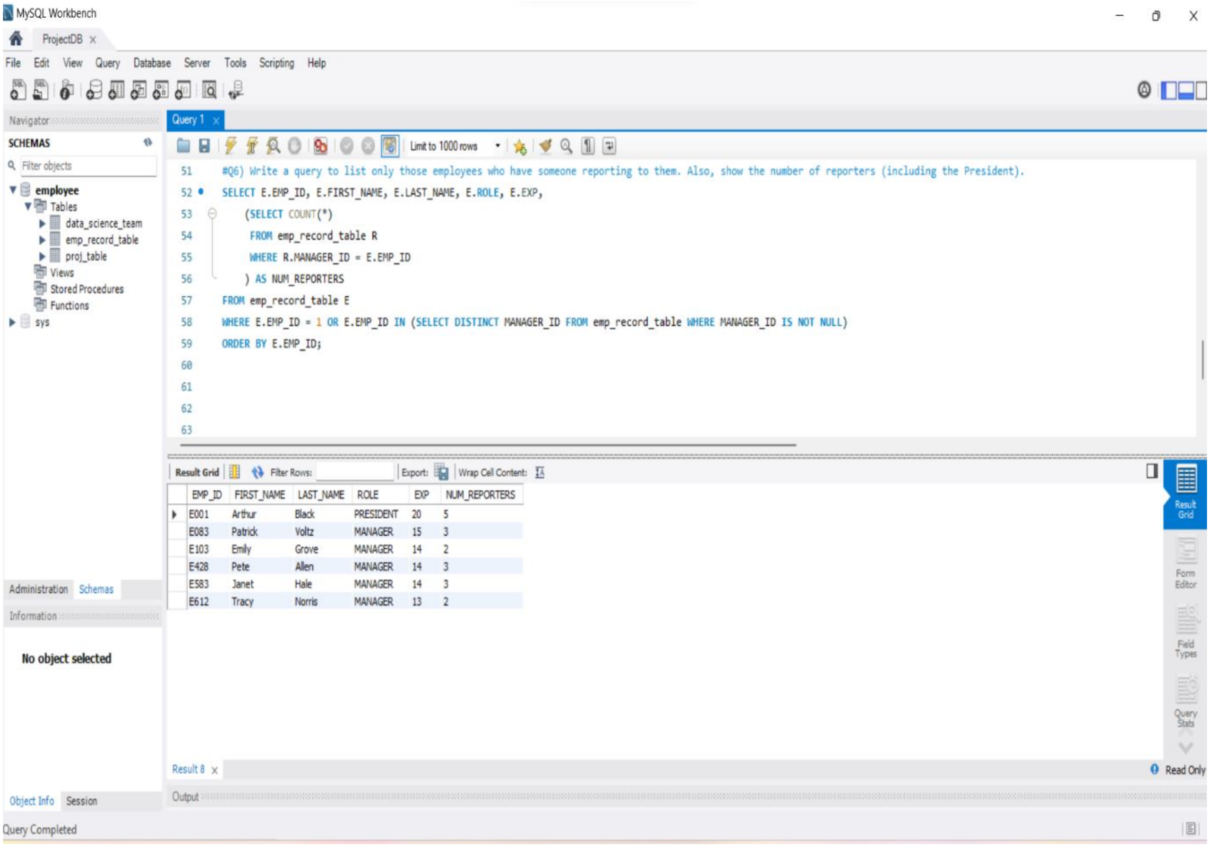
WHERE R.MANAGER_ID = E.EMP_ID

) AS NUM_REPORTERS

FROM emp_record_table E

WHERE E.EMP_ID = 1 OR E.EMP_ID IN (SELECT DISTINCT MANAGER_ID FROM emp_record_table WHERE MANAGER_ID IS NOT NULL)

ORDER BY E.EMP_ID;



```
#Q6) Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).
SELECT E.EMP_ID, E.FIRST_NAME, E.LAST_NAME, E.ROLE, E.EXP,
    (SELECT COUNT(*)
     FROM emp_record_table R
     WHERE R.MANAGER_ID = E.EMP_ID
    ) AS NUM_REPORTERS
FROM emp_record_table E
WHERE E.EMP_ID = 1 OR E.EMP_ID IN (SELECT DISTINCT MANAGER_ID FROM emp_record_table WHERE MANAGER_ID IS NOT NULL)
ORDER BY E.EMP_ID;
```

| EMP_ID | FIRST_NAME | LAST_NAME | ROLE | EXP | NUM_REPORTERS |
|---|---|---|---|---|---|
| E001 | Arthur | Black | PRESIDENT | 20 | 5 |
| E083 | Patrick | Voltz | MANAGER | 15 | 3 |
| E103 | Emily | Grove | MANAGER | 14 | 2 |
| E428 | Pete | Allen | MANAGER | 14 | 3 |
| E583 | Janet | Hale | MANAGER | 14 | 3 |
| E612 | Tracy | Norris | MANAGER | 13 | 2 |

**Q7) Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.**

**Ans.** SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT

FROM emp_record_table

WHERE DEPT = 'HEALTHCARE'

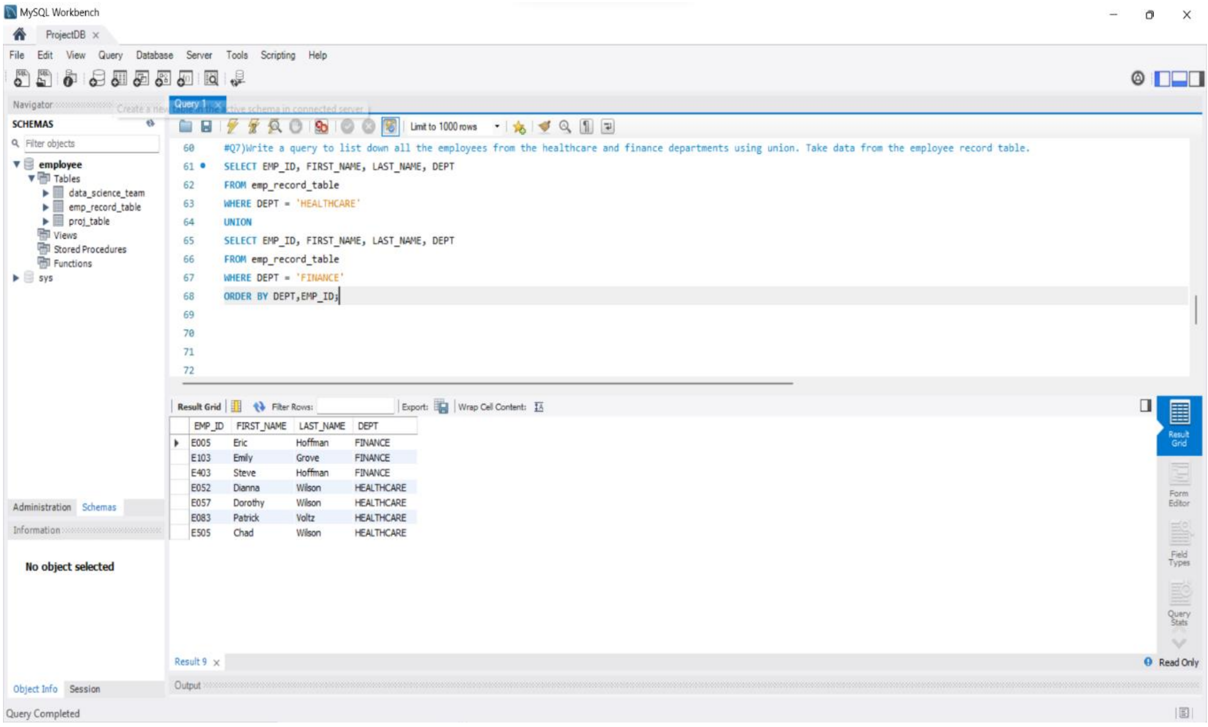
UNION

SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT

FROM emp_record_table

WHERE DEPT = 'FINANCE'

ORDER BY DEPT,EMP_ID;



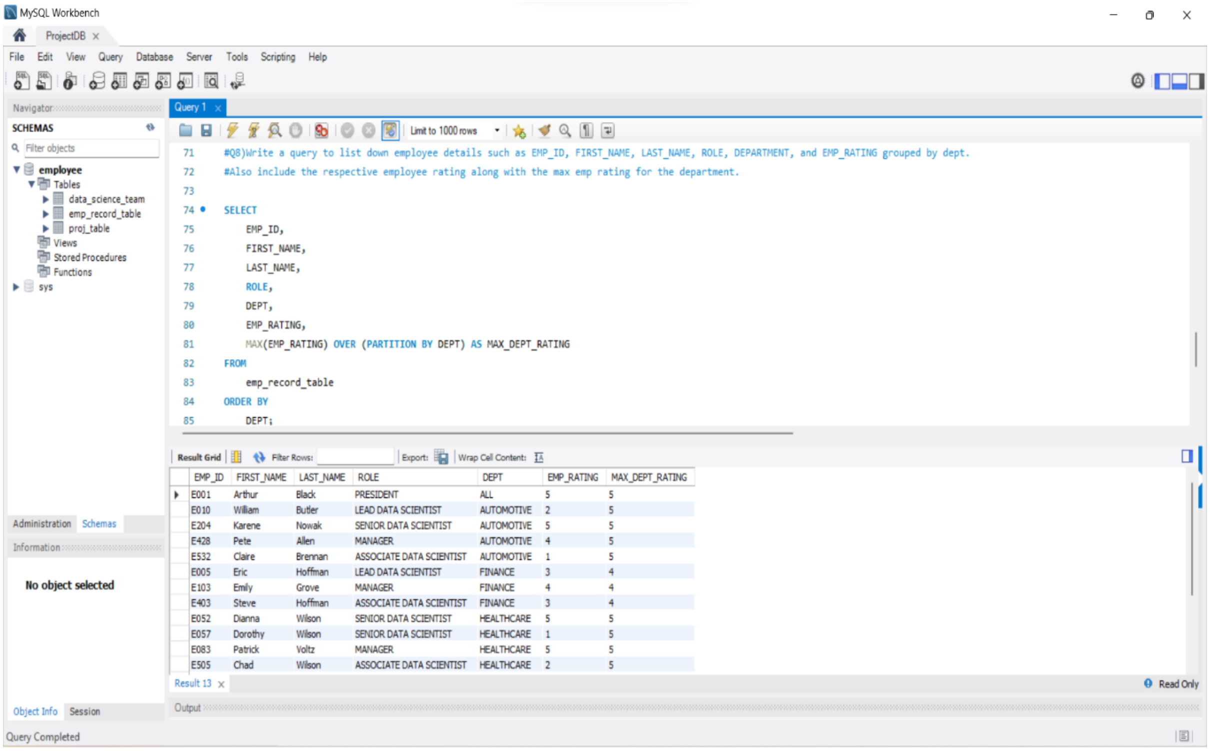
```
#Q7)Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.
SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT
FROM emp_record_table
WHERE DEPT = 'HEALTHCARE'
UNION
SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT
FROM emp_record_table
WHERE DEPT = 'FINANCE'
ORDER BY DEPT,EMP_ID;
```

| EMP_ID | FIRST_NAME | LAST_NAME | DEPT |
|---|---|---|---|
| E005 | Eric | Hoffman | FINANCE |
| E103 | Emily | Grove | FINANCE |
| E403 | Steve | Hoffman | FINANCE |
| E052 | Dianna | Wilson | HEALTHCARE |
| E057 | Dorothy | Wilson | HEALTHCARE |
| E083 | Patrick | Voltz | HEALTHCARE |
| E505 | Chad | Wilson | HEALTHCARE |

**Q8) 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.**

**Ans.**

```
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    ROLE,
    DEPT,
    EMP_RATING,
    MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MAX_DEPT_RATING
FROM
    emp_record_table
ORDER BY
    DEPT;
```



```
#Q8)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.

SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    ROLE,
    DEPT,
    EMP_RATING,
    MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MAX_DEPT_RATING
FROM
    emp_record_table
ORDER BY
    DEPT;
```

| EMP_ID | FIRST_NAME | LAST_NAME | ROLE | DEPT | EMP_RATING | MAX_DEPT_RATING |
|---|---|---|---|---|---|---|
| E001 | Arthur | Black | PRESIDENT | ALL | 5 | 5 |
| E010 | William | Butler | LEAD DATA SCIENTIST | AUTOMOTIVE | 2 | 5 |
| E204 | Karene | Nowak | SENIOR DATA SCIENTIST | AUTOMOTIVE | 5 | 5 |
| E428 | Pete | Allen | MANAGER | AUTOMOTIVE | 4 | 5 |
| E532 | Claire | Brennan | ASSOCIATE DATA SCIENTIST | AUTOMOTIVE | 1 | 5 |
| E005 | Eric | Hoffman | LEAD DATA SCIENTIST | FINANCE | 3 | 4 |
| E103 | Emily | Grove | MANAGER | FINANCE | 4 | 4 |
| E403 | Steve | Hoffman | ASSOCIATE DATA SCIENTIST | FINANCE | 3 | 4 |
| E052 | Dianna | Wilson | SENIOR DATA SCIENTIST | HEALTHCARE | 5 | 5 |
| E057 | Dorothy | Wilson | SENIOR DATA SCIENTIST | HEALTHCARE | 1 | 5 |
| E083 | Patrick | Voltz | MANAGER | HEALTHCARE | 5 | 5 |
| E505 | Chad | Wilson | ASSOCIATE DATA SCIENTIST | HEALTHCARE | 2 | 5 |

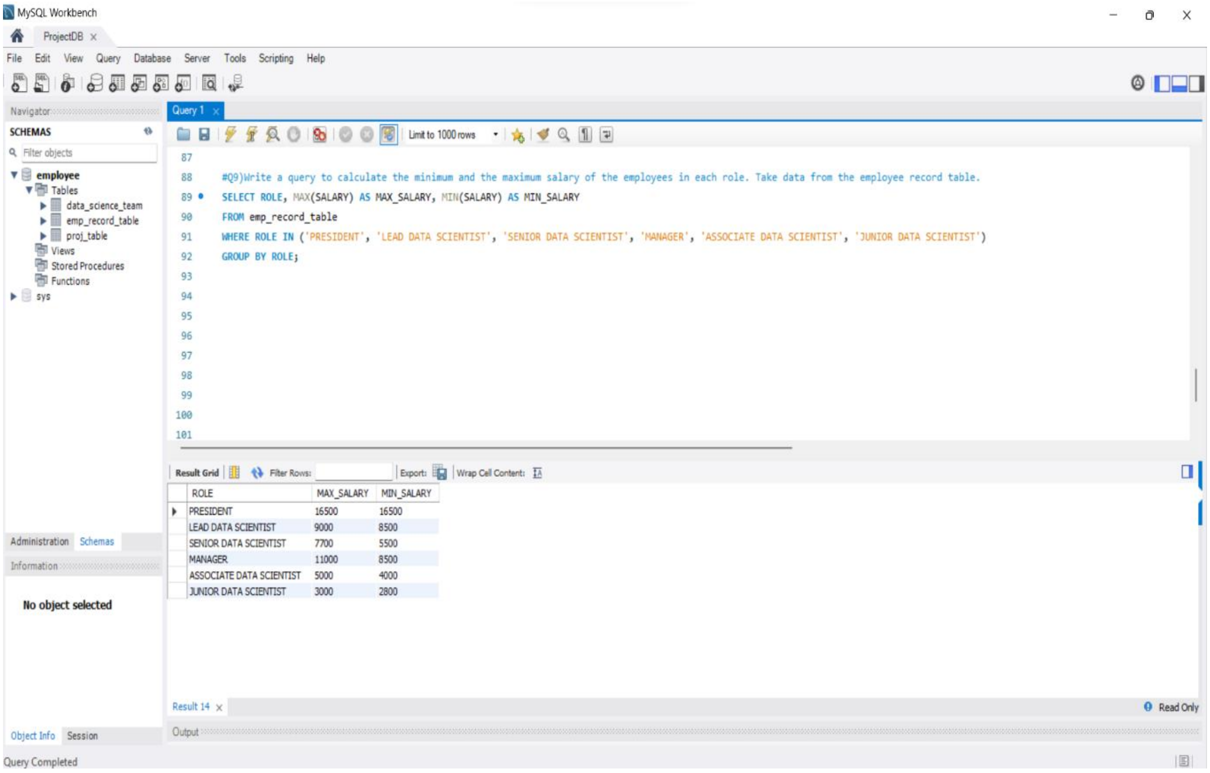
**Q9) Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.**

**Ans.** SELECT ROLE, MAX(SALARY) AS MAX_SALARY, MIN(SALARY) AS 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;



```
#Q9)Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.
SELECT ROLE, MAX(SALARY) AS MAX_SALARY, MIN(SALARY) AS 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;
```

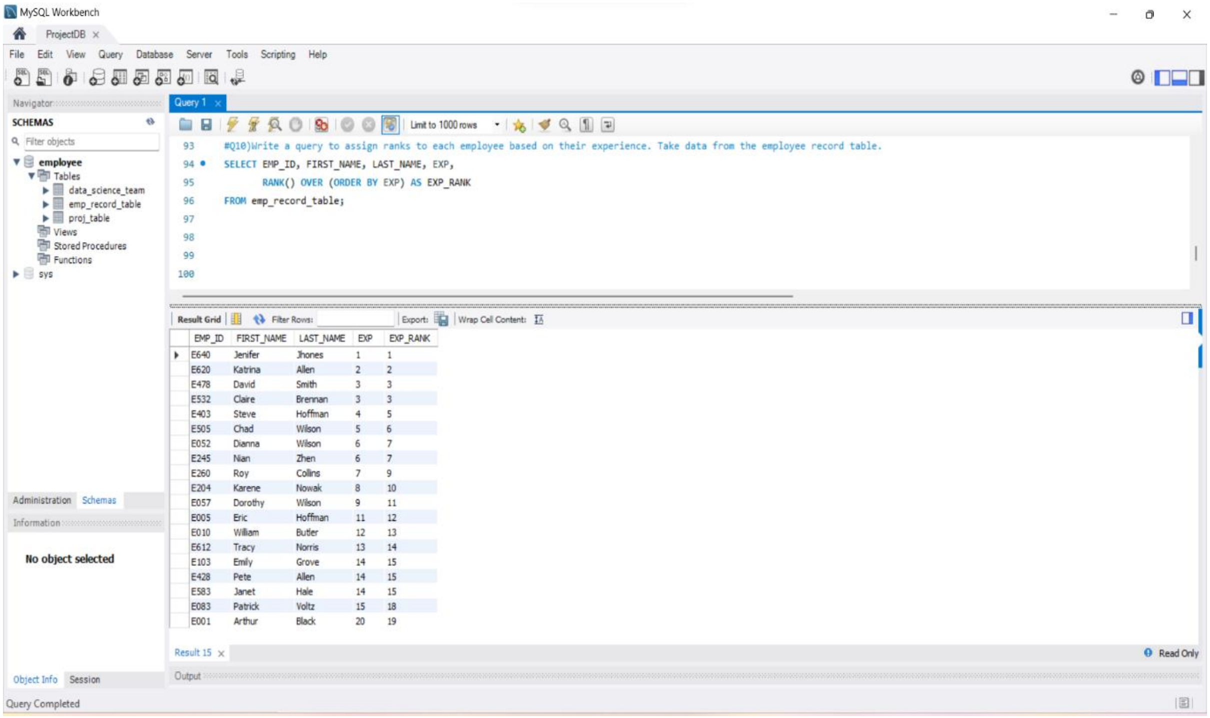
| ROLE | MAX_SALARY | MIN_SALARY |
|---|---|---|
| PRESIDENT | 16500 | 16500 |
| LEAD DATA SCIENTIST | 9000 | 8500 |
| SENIOR DATA SCIENTIST | 7700 | 5500 |
| MANAGER | 11000 | 8500 |
| ASSOCIATE DATA SCIENTIST | 5000 | 4000 |
| JUNIOR DATA SCIENTIST | 3000 | 2800 |

**Q10) Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.**

**Ans.** SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP,

RANK() OVER (ORDER BY EXP) AS EXP_RANK

FROM emp_record_table;



```
#Q10)Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.
SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP,
        RANK() OVER (ORDER BY EXP) AS EXP_RANK
FROM emp_record_table;
```

| EMP_ID | FIRST_NAME | LAST_NAME | EXP | EXP_RANK |
|---|---|---|---|---|
| E640 | Jenifer | Jhones | 1 | 1 |
| E620 | Katrina | Allen | 2 | 2 |
| E478 | David | Smith | 3 | 3 |
| E532 | Claire | Brennan | 3 | 3 |
| E403 | Steve | Hoffman | 4 | 5 |
| E505 | Chad | Wilson | 5 | 6 |
| E052 | Dianna | Wilson | 6 | 7 |
| E245 | Nian | Zhen | 6 | 7 |
| E260 | Roy | Collins | 7 | 9 |
| E204 | Karene | Nowak | 8 | 10 |
| E057 | Dorothy | Wilson | 9 | 11 |
| E005 | Eric | Hoffman | 11 | 12 |
| E010 | William | Butler | 12 | 13 |
| E612 | Tracy | Norris | 13 | 14 |
| E103 | Emily | Grove | 14 | 15 |
| E428 | Pete | Allen | 14 | 15 |
| E583 | Janet | Hale | 14 | 15 |
| E083 | Patrick | Voltz | 15 | 18 |
| E001 | Arthur | Black | 20 | 19 |

**Q11) 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.**

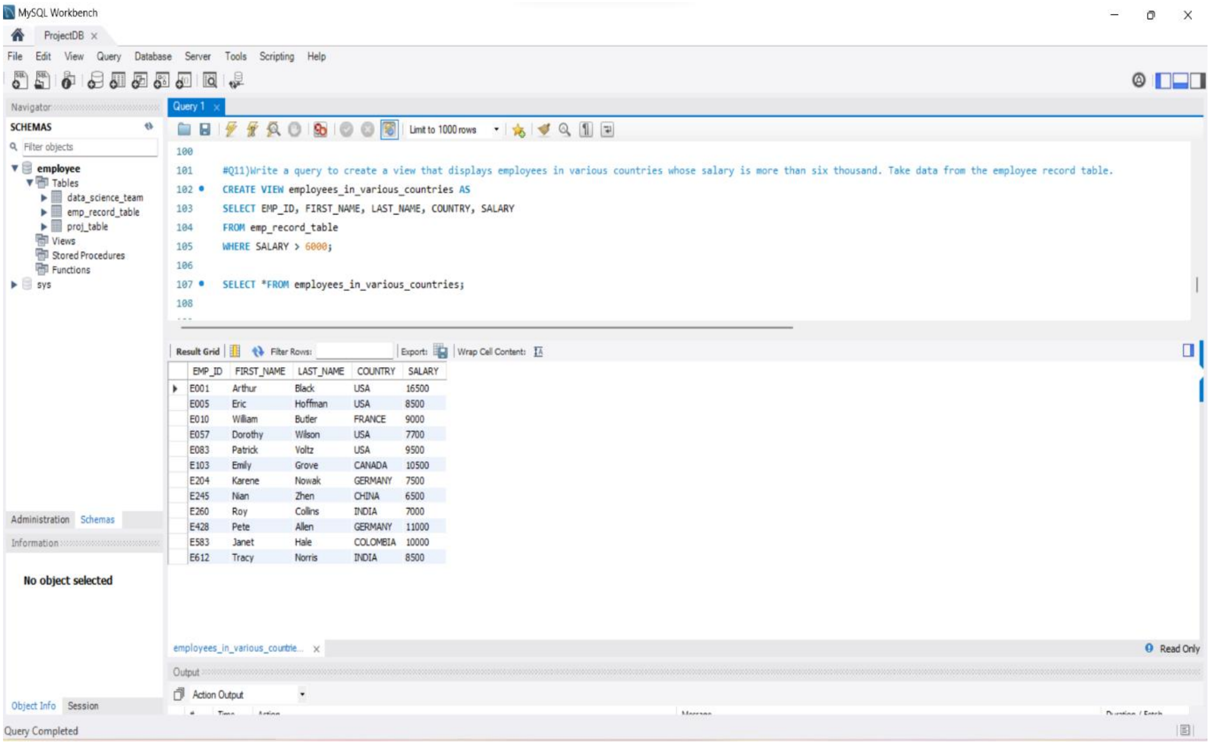
**Ans.** CREATE VIEW employees_in_various_countries AS

SELECT EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY

FROM emp_record_table

WHERE SALARY > 6000;

SELECT *FROM employees_in_various_countries;



```
#Q11)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.
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;
```

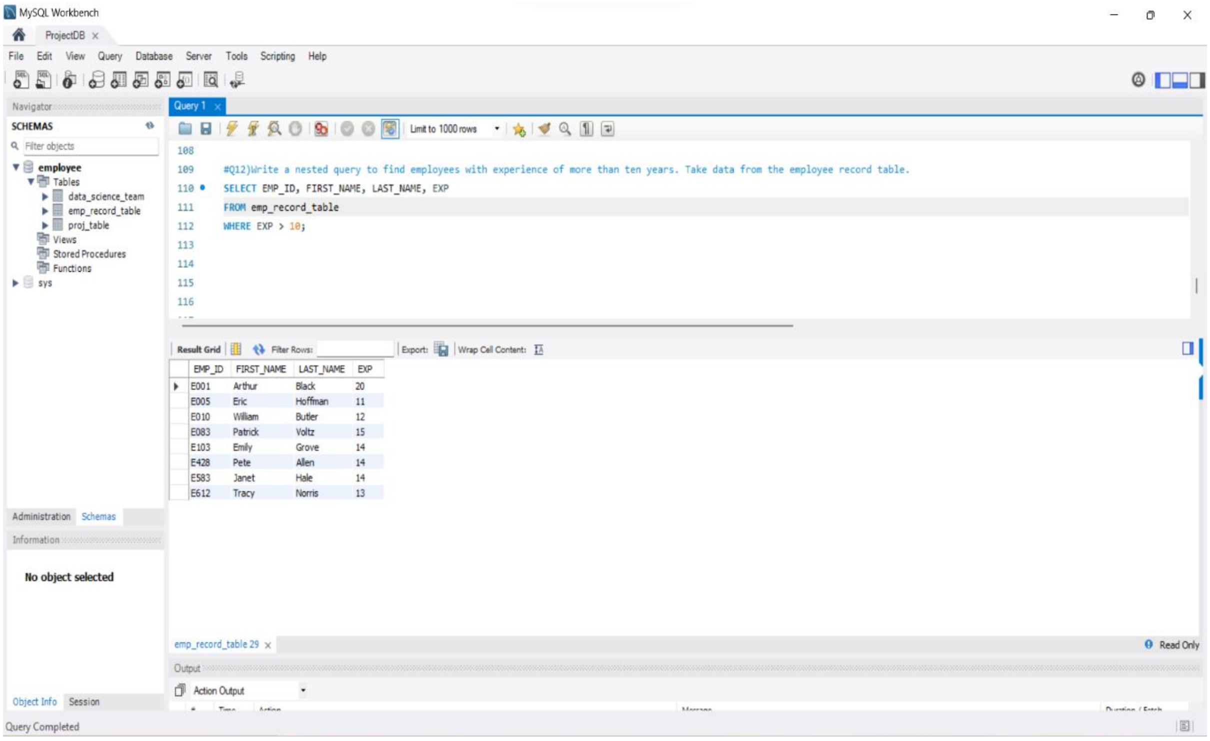
| EMP_ID | FIRST_NAME | LAST_NAME | COUNTRY | SALARY |
|---|---|---|---|---|
| E001 | Arthur | Black | USA | 16500 |
| E005 | Eric | Hoffman | USA | 8500 |
| E010 | William | Butler | FRANCE | 9000 |
| E057 | Dorothy | Wilson | USA | 7700 |
| E083 | Patrick | Voltz | USA | 9500 |
| E103 | Emily | Grove | CANADA | 10500 |
| E204 | Karene | Nowak | GERMANY | 7500 |
| E245 | Nian | Zhen | CHINA | 6500 |
| E260 | Roy | Collins | INDIA | 7000 |
| E428 | Pete | Allen | GERMANY | 11000 |
| E583 | Janet | Hale | COLOMBIA | 10000 |
| E612 | Tracy | Norris | INDIA | 8500 |

**Q12) Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.**

**Ans.** SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP

FROM emp_record_table

WHERE EXP > 10;



```
#Q12)Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.
SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP
FROM emp_record_table
WHERE EXP > 10;
```

| EMP_ID | FIRST_NAME | LAST_NAME | EXP |
|---|---|---|---|
| E001 | Arthur | Black | 20 |
| E005 | Eric | Hoffman | 11 |
| E010 | William | Butler | 12 |
| E083 | Patrick | Voltz | 15 |
| E103 | Emily | Grove | 14 |
| E428 | Pete | Allen | 14 |
| E583 | Janet | Hale | 14 |
| E612 | Tracy | Norris | 13 |

**Q13) 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.**

**Ans.** DELIMITER &&

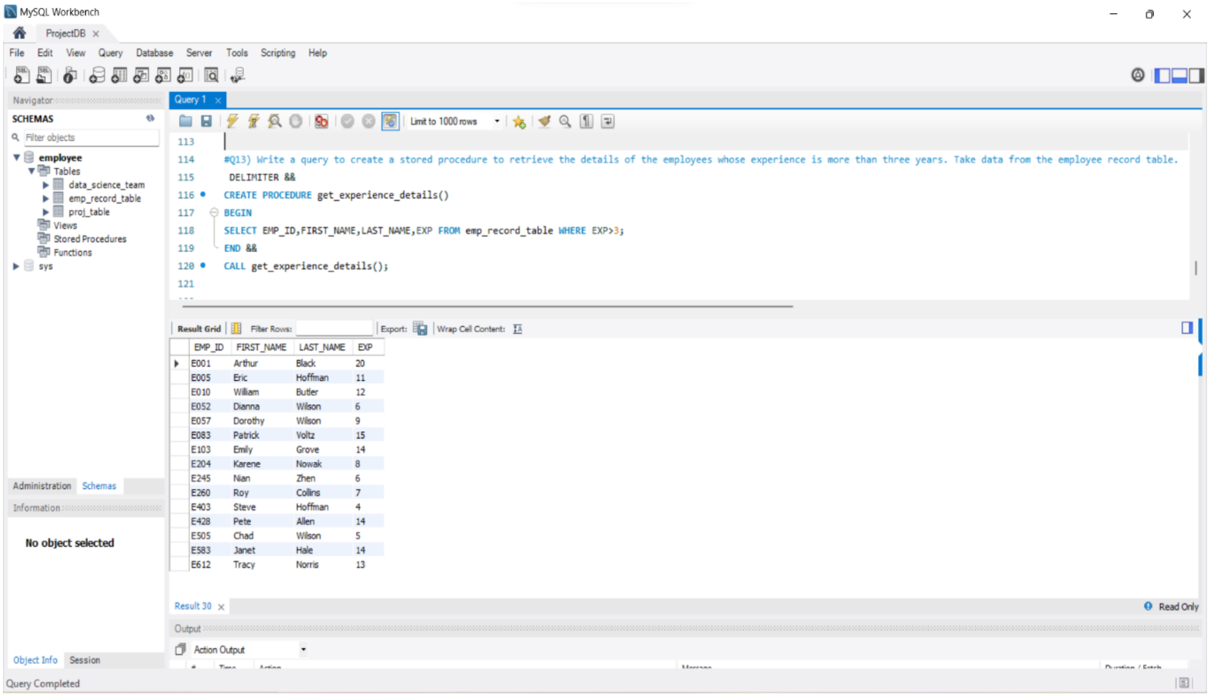
CREATE PROCEDURE get_experience_details()

BEGIN

SELECT EMP_ID,FIRST_NAME,LAST_NAME,EXP FROM emp_record_table WHERE EXP>3;

END &&

CALL get_experience_details();



```
#Q13) 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.
 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();
```

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

**Q14) 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',**

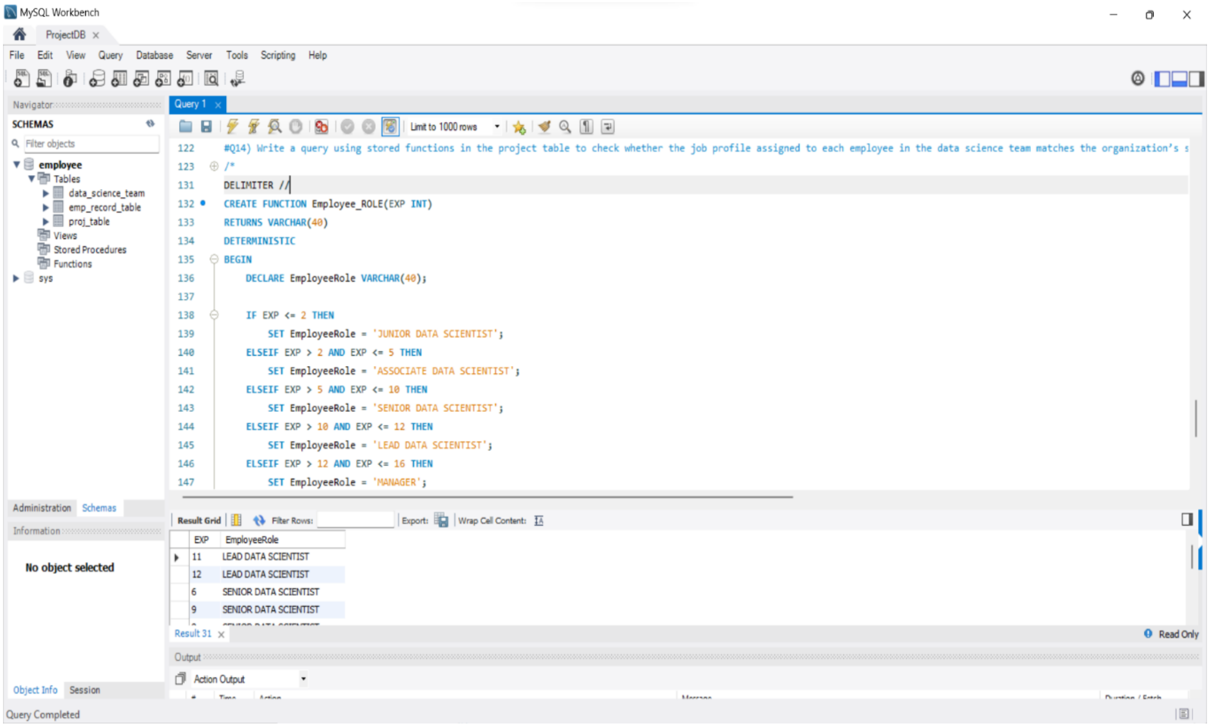
**For an employee with the experience of 5 to 10 years assign 'SENIOR DATA SCIENTIST',**

**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'.**

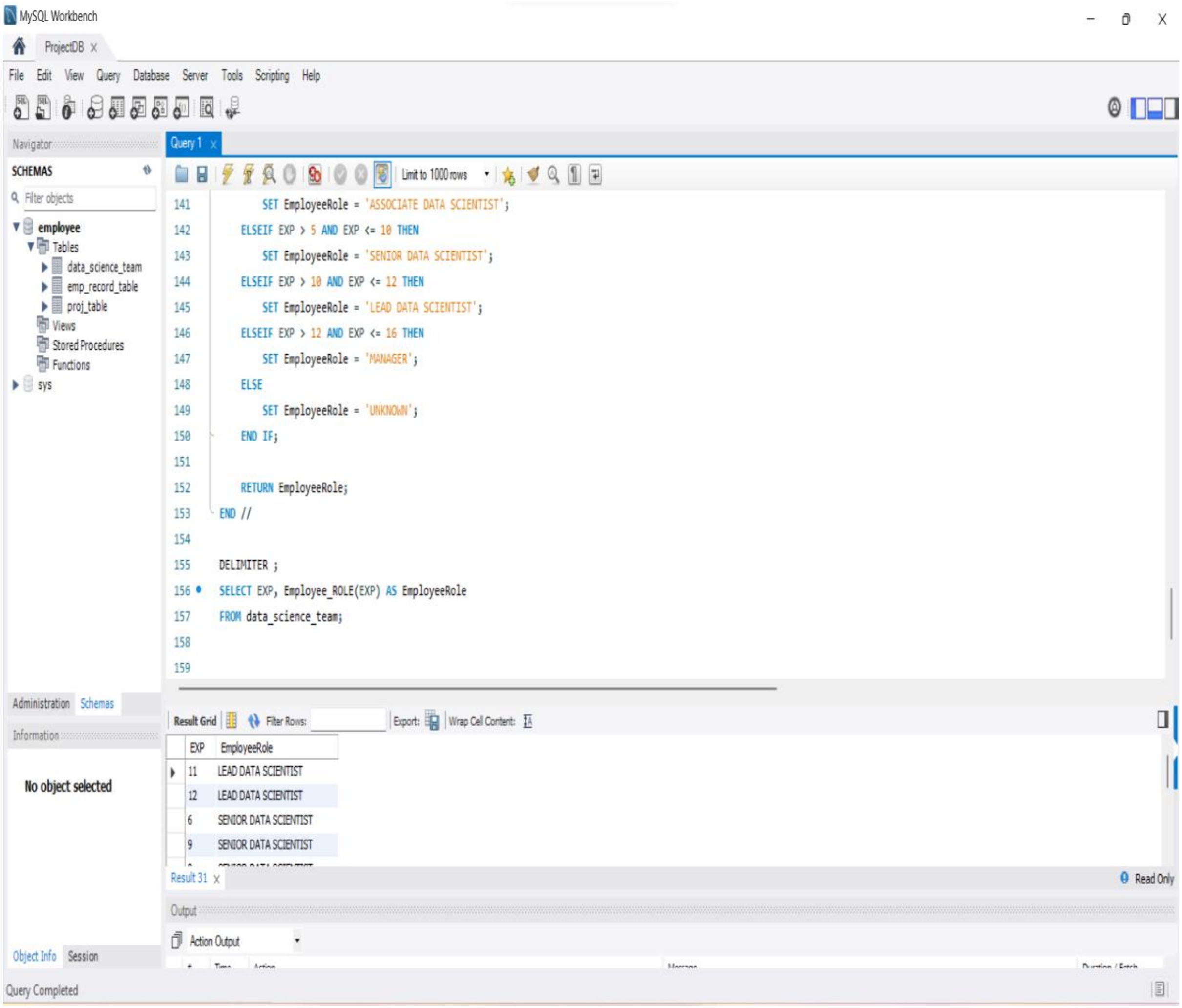
**Ans.** DELIMITER //

```
CREATE FUNCTION Employee_ROLE(EXP INT)

RETURNS VARCHAR(40)

DETERMINISTIC

BEGIN

  DECLARE EmployeeRole VARCHAR(40);


  IF EXP <= 2 THEN

    SET EmployeeRole = 'JUNIOR DATA SCIENTIST';

  ELSEIF EXP > 2 AND EXP <= 5 THEN

    SET EmployeeRole = 'ASSOCIATE DATA SCIENTIST';
    ELSEIF EXP > 5 AND EXP <= 10 THEN
        SET EmployeeRole = 'SENIOR DATA SCIENTIST';
    ELSEIF EXP > 10 AND EXP <= 12 THEN
        SET EmployeeRole = 'LEAD DATA SCIENTIST';
    ELSEIF EXP > 12 AND EXP <= 16 THEN
        SET EmployeeRole = 'MANAGER';
    ELSE
        SET EmployeeRole = 'UNKNOWN';
    END IF;

    RETURN EmployeeRole;
END //

DELIMITER ;
SELECT EXP, Employee_ROLE(EXP) AS EmployeeRole
FROM data_science_team;
```



```
#Q14) 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 s
/*
DELIMITER //
CREATE FUNCTION Employee_ROLE(EXP INT)
RETURNS VARCHAR(40)
DETERMINISTIC
BEGIN
    DECLARE EmployeeRole VARCHAR(40);

    IF EXP <= 2 THEN
        SET EmployeeRole = 'JUNIOR DATA SCIENTIST';
    ELSEIF EXP > 2 AND EXP <= 5 THEN
        SET EmployeeRole = 'ASSOCIATE DATA SCIENTIST';
    ELSEIF EXP > 5 AND EXP <= 10 THEN
        SET EmployeeRole = 'SENIOR DATA SCIENTIST';
    ELSEIF EXP > 10 AND EXP <= 12 THEN
        SET EmployeeRole = 'LEAD DATA SCIENTIST';
    ELSEIF EXP > 12 AND EXP <= 16 THEN
        SET EmployeeRole = 'MANAGER';
```

| EXP | EmployeeRole |
|---|---|
| 11 | LEAD DATA SCIENTIST |
| 12 | LEAD DATA SCIENTIST |
| 6 | SENIOR DATA SCIENTIST |
| 9 | SENIOR DATA SCIENTIST |



```
        SET EmployeeRole = 'ASSOCIATE DATA SCIENTIST';
    ELSEIF EXP > 5 AND EXP <= 10 THEN
        SET EmployeeRole = 'SENIOR DATA SCIENTIST';
    ELSEIF EXP > 10 AND EXP <= 12 THEN
        SET EmployeeRole = 'LEAD DATA SCIENTIST';
    ELSEIF EXP > 12 AND EXP <= 16 THEN
        SET EmployeeRole = 'MANAGER';
    ELSE
        SET EmployeeRole = 'UNKNOWN';
    END IF;

    RETURN EmployeeRole;
END //

DELIMITER ;
SELECT EXP, Employee_ROLE(EXP) AS EmployeeRole
FROM data_science_team;
```

| EXP | EmployeeRole |
|---|---|
| 11 | LEAD DATA SCIENTIST |
| 12 | LEAD DATA SCIENTIST |
| 6 | SENIOR DATA SCIENTIST |
| 9 | SENIOR DATA SCIENTIST |

**Q15) 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.**

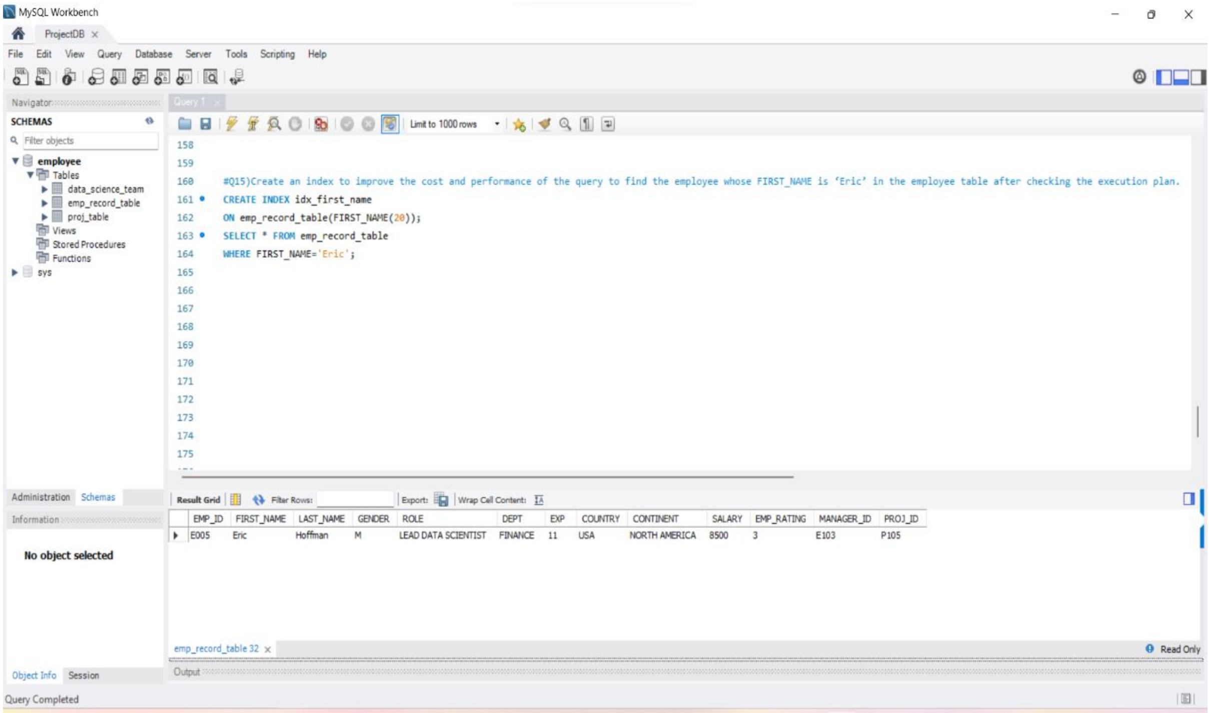
**Ans.**

CREATE INDEX idx_first_name

ON emp_record_table(FIRST_NAME(20));

SELECT * FROM emp_record_table

WHERE FIRST_NAME='Eric';



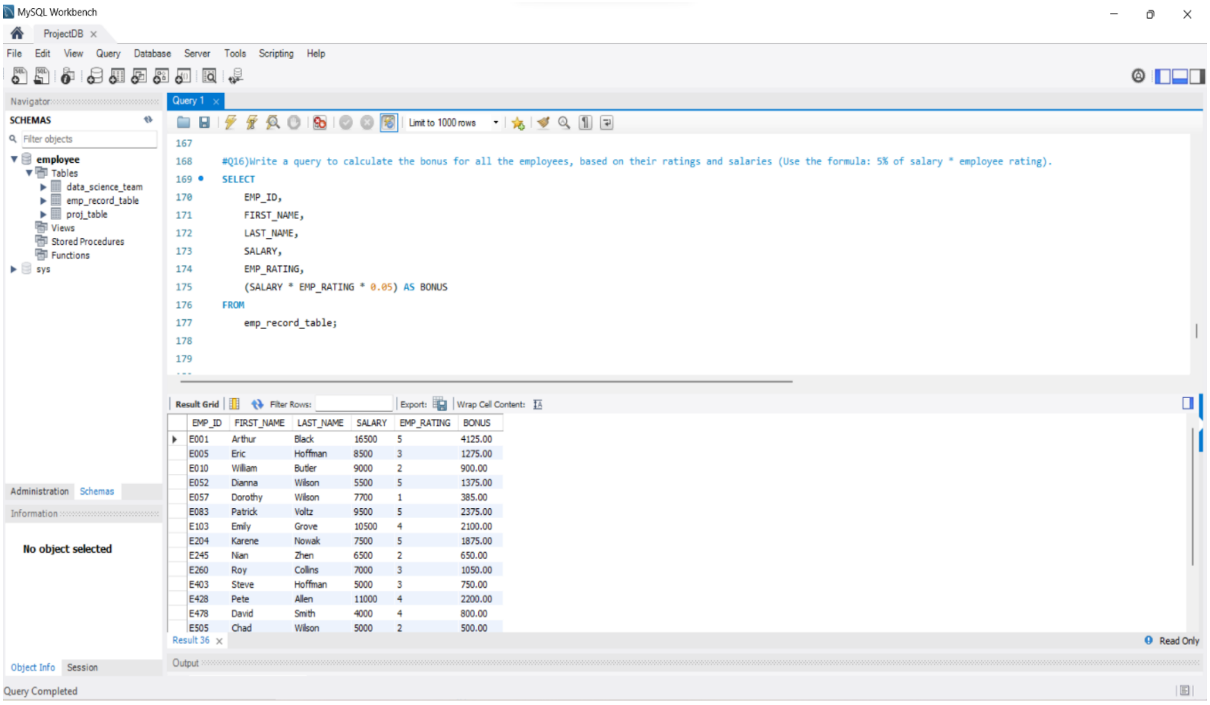
```
#Q15)Create an index to improve the cost and performance of the query to find the employee whose FIRST_NAME is 'Eric' in the employee table after checking the execution plan.
CREATE INDEX idx_first_name
ON emp_record_table(FIRST_NAME(20));
SELECT * FROM emp_record_table
WHERE FIRST_NAME='Eric';
```

| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | ROLE | DEPT | EXP | COUNTRY | CONTINENT | SALARY | EMP_RATING | MANAGER_ID | PROJ_ID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E005 | Eric | Hoffman | M | LEAD DATA SCIENTIST | FINANCE | 11 | USA | NORTH AMERICA | 8500 | 3 | E103 | P105 |

**Q16) 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).**

**Ans.**

```
SELECT
   EMP_ID,
   FIRST_NAME,
   LAST_NAME,
   SALARY,
   EMP_RATING,
   (SALARY * EMP_RATING * 0.05) AS BONUS
FROM
   emp_record_table;
```



```
#Q16)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).
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    SALARY,
    EMP_RATING,
    (SALARY * EMP_RATING * 0.05) AS BONUS
FROM
    emp_record_table;
```

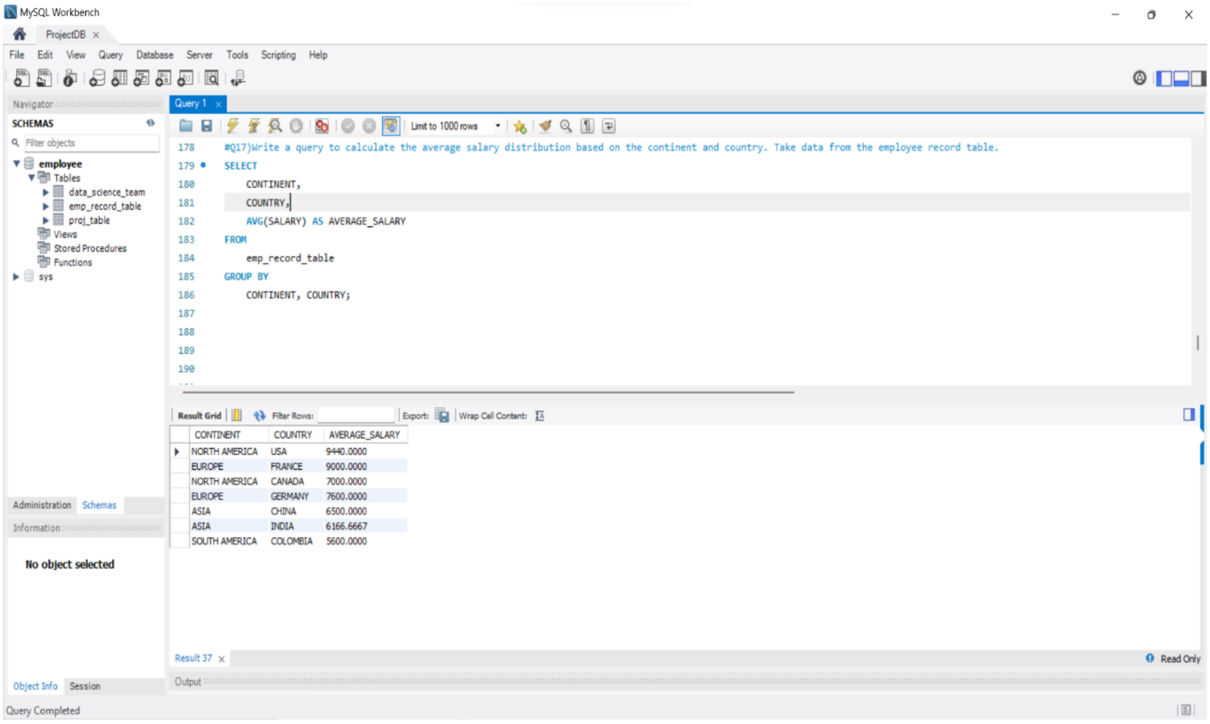
| EMP_ID | FIRST_NAME | LAST_NAME | SALARY | EMP_RATING | BONUS |
|---|---|---|---|---|---|
| E001 | Arthur | Black | 16500 | 5 | 4125.00 |
| E005 | Eric | Hoffman | 8500 | 3 | 1275.00 |
| E010 | William | Butler | 9000 | 2 | 900.00 |
| E052 | Dianna | Wilson | 5500 | 5 | 1375.00 |
| E057 | Dorothy | Wilson | 7700 | 1 | 385.00 |
| E083 | Patrick | Voltz | 9500 | 5 | 2375.00 |
| E103 | Emily | Grove | 10500 | 4 | 2100.00 |
| E204 | Karene | Nowak | 7500 | 5 | 1875.00 |
| E245 | Nian | Zhen | 6500 | 2 | 650.00 |
| E260 | Roy | Collins | 7000 | 3 | 1050.00 |
| E403 | Steve | Hoffman | 5000 | 3 | 750.00 |
| E428 | Pete | Allen | 11000 | 4 | 2200.00 |
| E478 | David | Smith | 4000 | 4 | 800.00 |
| E505 | Chad | Wilson | 5000 | 2 | 500.00 |

Query Completed

**Q17) Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.**

**Ans.**

```
 SELECT
    CONTINENT,
    COUNTRY,
    AVG(SALARY) AS AVERAGE_SALARY
FROM
    emp_record_table
GROUP BY
    CONTINENT, COUNTRY;
```



```
#Q17)Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.
SELECT
    CONTINENT,
    COUNTRY,
    AVG(SALARY) AS AVERAGE_SALARY
FROM
    emp_record_table
GROUP BY
    CONTINENT, COUNTRY;
```

| CONTINENT | COUNTRY | AVERAGE_SALARY |
|---|---|---|
| NORTH AMERICA | USA | 9440.0000 |
| EUROPE | FRANCE | 9000.0000 |
| NORTH AMERICA | CANADA | 7000.0000 |
| EUROPE | GERMANY | 7600.0000 |
| ASIA | CHINA | 6500.0000 |
| ASIA | INDIA | 6166.6667 |
| SOUTH AMERICA | COLOMBIA | 5600.0000 |