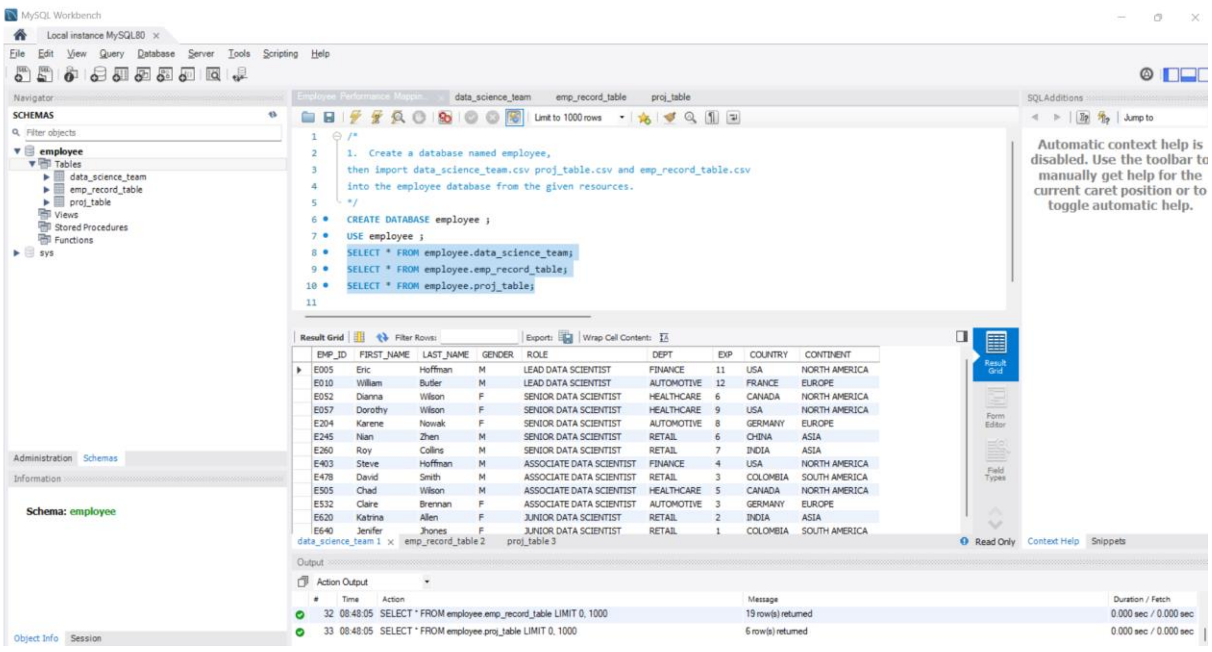
ScienceQtech Employee Performance Mapping

Introduction:
This project is based on the Employee Performance Mapping case study. The objective is to analyze employee records, project details, and data science team information using SQL queries. The tasks involve creating an ER diagram, writing queries for filtering, aggregation, and ranking, as well as implementing advanced concepts like stored procedures, stored functions, indexing, and views.
The project demonstrates how SQL can be applied in real-world scenarios to generate insights for HR departments, improve employee performance evaluation, and optimize organizational decisions.

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, we should find the maximum salary of the employees and ensure that all jobs are meeting the organization's profile standard. we 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

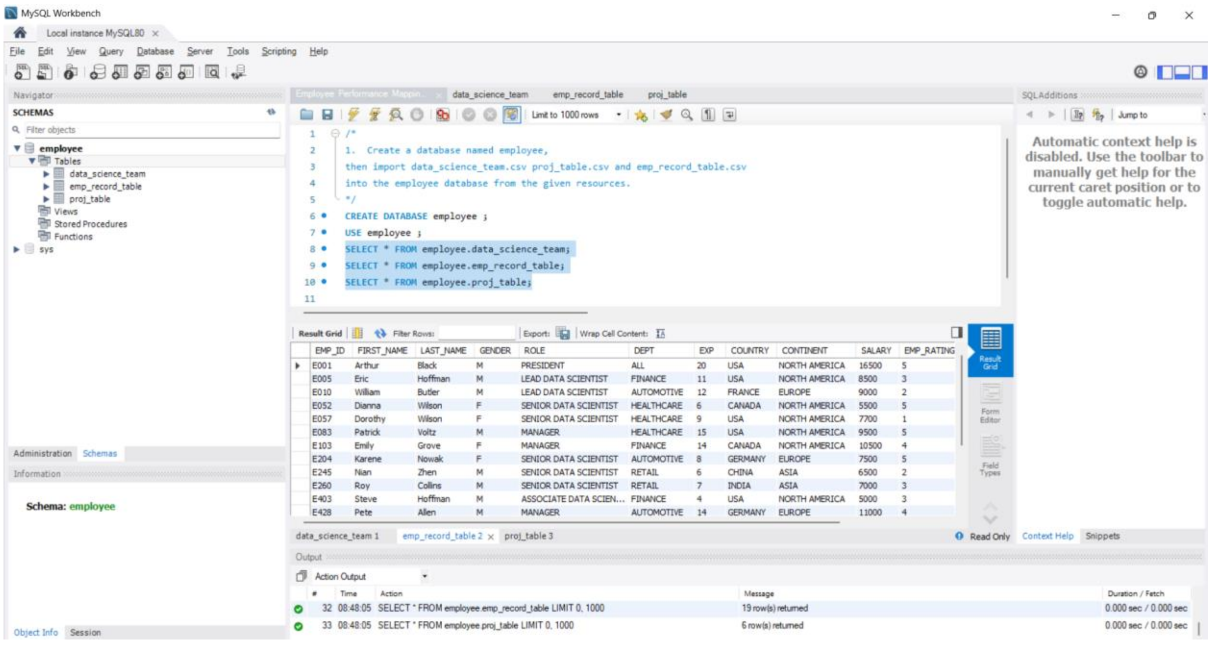
1. Create a database named *employee*, then import **data_science_team.csv proj_table.csv** and **emp_record_table.csv** into the **employee** database from the given resources.



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

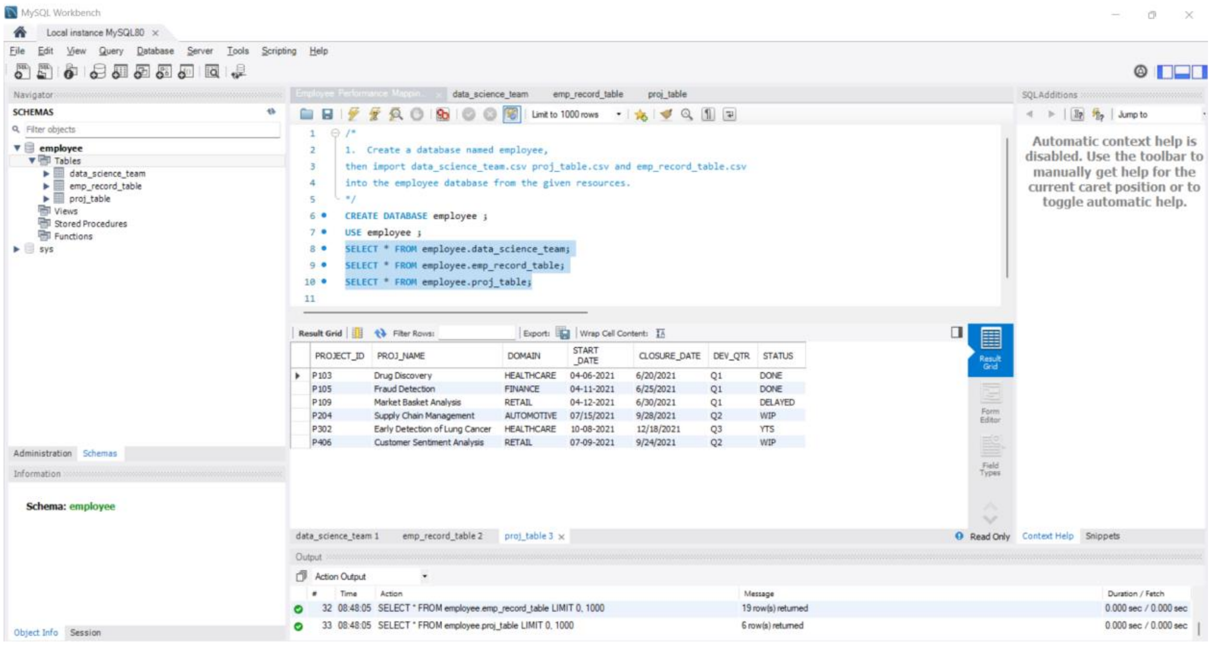
```
#   Time      Action                                                  Message            Duration / Fetch
32  08:48:05  SELECT * FROM employee.emp_record_table LIMIT 0, 1000   19 row(s) returned  0.000 sec / 0.000 sec
33  08:48:05  SELECT * FROM employee.proj_table LIMIT 0, 1000         6 row(s) returned   0.000 sec / 0.000 sec
```



```sql
/*
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.
*/
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	SALARY	EMP_RATING
E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500	5
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4
E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5
E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500	2
E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIEN...	FINANCE	4	USA	NORTH AMERICA	5000	3
E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4

```
#   Time      Action                                                 Message              Duration / Fetch
32  08:48:05  SELECT * FROM employee.emp_record_table LIMIT 0, 1000  19 row(s) returned   0.000 sec / 0.000 sec
33  08:48:05  SELECT * FROM employee.proj_table LIMIT 0, 1000        6 row(s) returned    0.000 sec / 0.000 sec
```

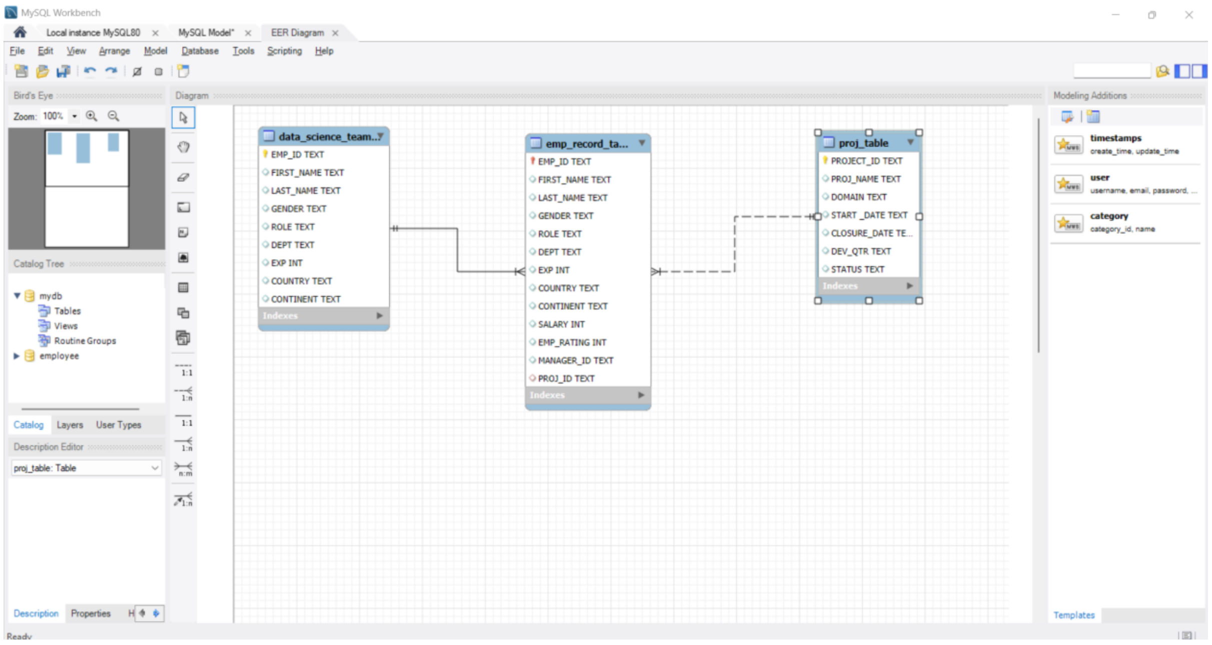


```sql
/*
1.  Create a database named employee,
then import data_science_team.csv proj_table.csv and emp_record_table.csv
into the employee database from the given resources.
*/
CREATE DATABASE employee ;
USE employee ;
SELECT * FROM employee.data_science_team;
SELECT * FROM employee.emp_record_table;
SELECT * FROM employee.proj_table;
```

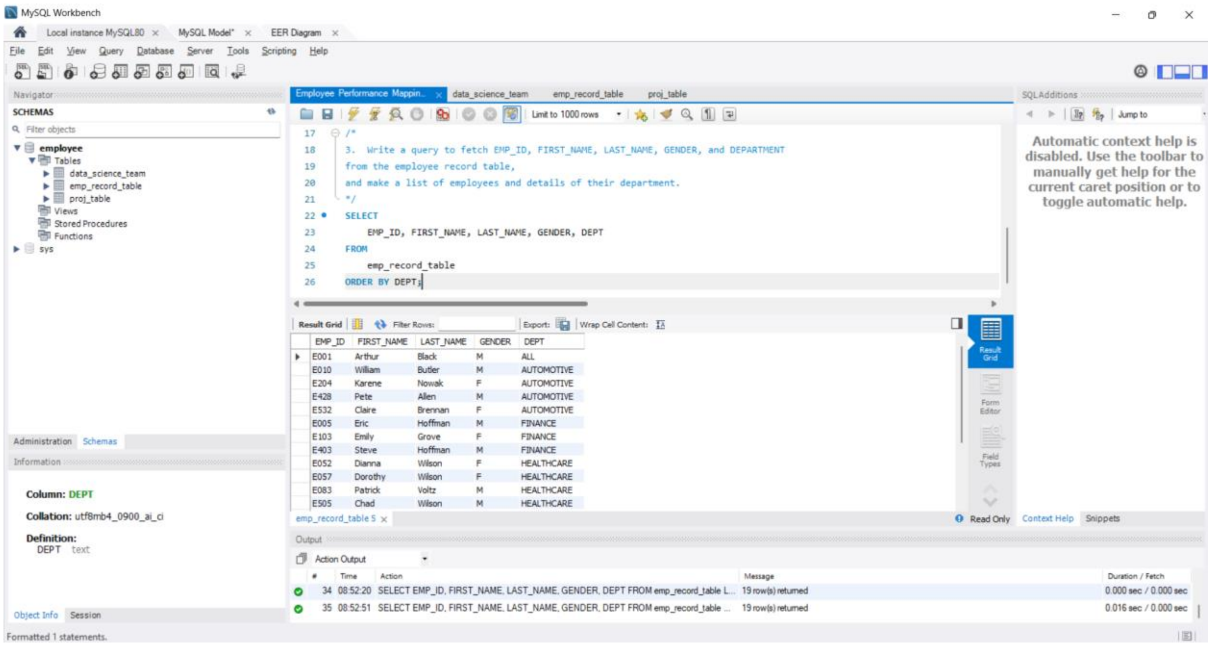
PROJECT_ID	PROJ_NAME	DOMAIN	START_DATE	CLOSURE_DATE	DEV_QTR	STATUS
P103	Drug Discovery	HEALTHCARE	04-06-2021	6/20/2021	Q1	DONE
P105	Fraud Detection	FINANCE	04-11-2021	6/25/2021	Q1	DONE
P109	Market Basket Analysis	RETAIL	04-12-2021	6/30/2021	Q1	DELAYED
P204	Supply Chain Management	AUTOMOTIVE	07/15/2021	9/28/2021	Q2	WIP
P302	Early Detection of Lung Cancer	HEALTHCARE	10-08-2021	12/18/2021	Q3	YTS
P406	Customer Sentiment Analysis	RETAIL	07-09-2021	9/24/2021	Q2	WIP

```
#   Time      Action                                                 Message              Duration / Fetch
32  08:48:05  SELECT * FROM employee.emp_record_table LIMIT 0, 1000  19 row(s) returned   0.000 sec / 0.000 sec
33  08:48:05  SELECT * FROM employee.proj_table LIMIT 0, 1000        6 row(s) returned    0.000 sec / 0.000 sec
```

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.



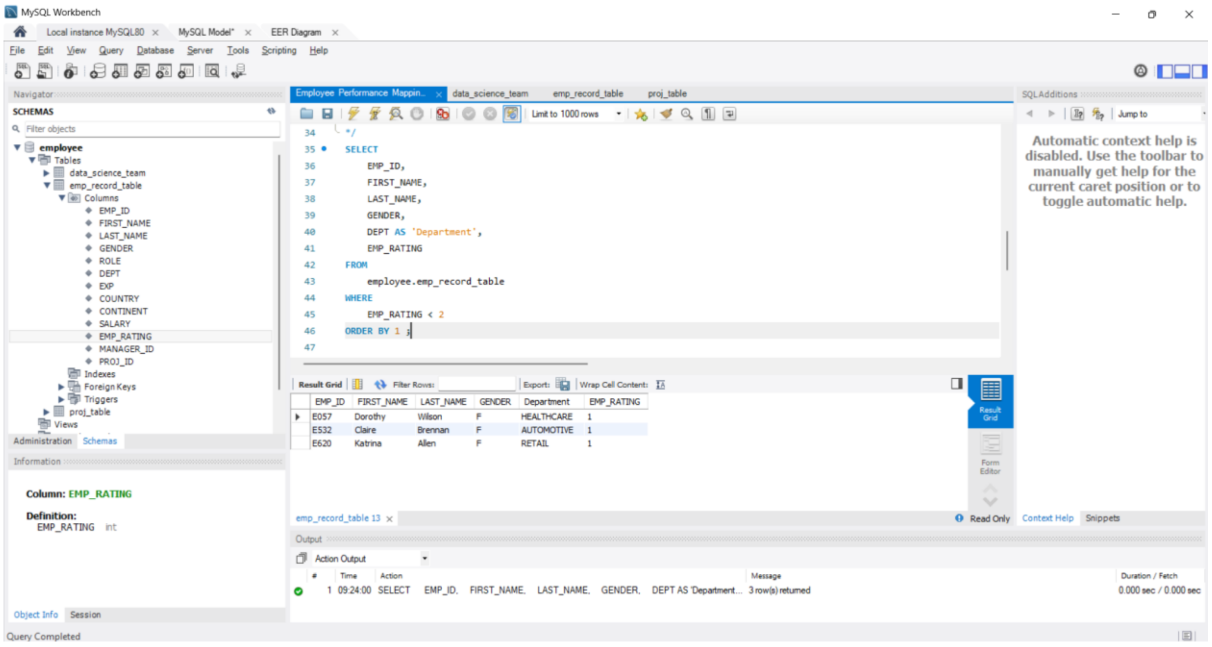
```
/*
3.  Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT
from the employee record table,
and make a list of employees and details of their department.
*/
SELECT
    EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM
    emp_record_table
ORDER BY DEPT;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E001	Arthur	Black	M	ALL
E010	William	Butler	M	AUTOMOTIVE
E204	Karene	Nowak	F	AUTOMOTIVE
E428	Pete	Allen	M	AUTOMOTIVE
E532	Claire	Brennan	F	AUTOMOTIVE
E005	Eric	Hoffman	M	FINANCE
E103	Emily	Grove	F	FINANCE
E403	Steve	Hoffman	M	FINANCE
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E505	Chad	Wilson	M	HEALTHCARE

#	Time	Action	Message	Duration / Fetch
34	08:52:20	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table L...	19 row(s) returned	0.000 sec / 0.000 sec
35	08:52:51	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table ...	19 row(s) returned	0.016 sec / 0.000 sec

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

Less than two:

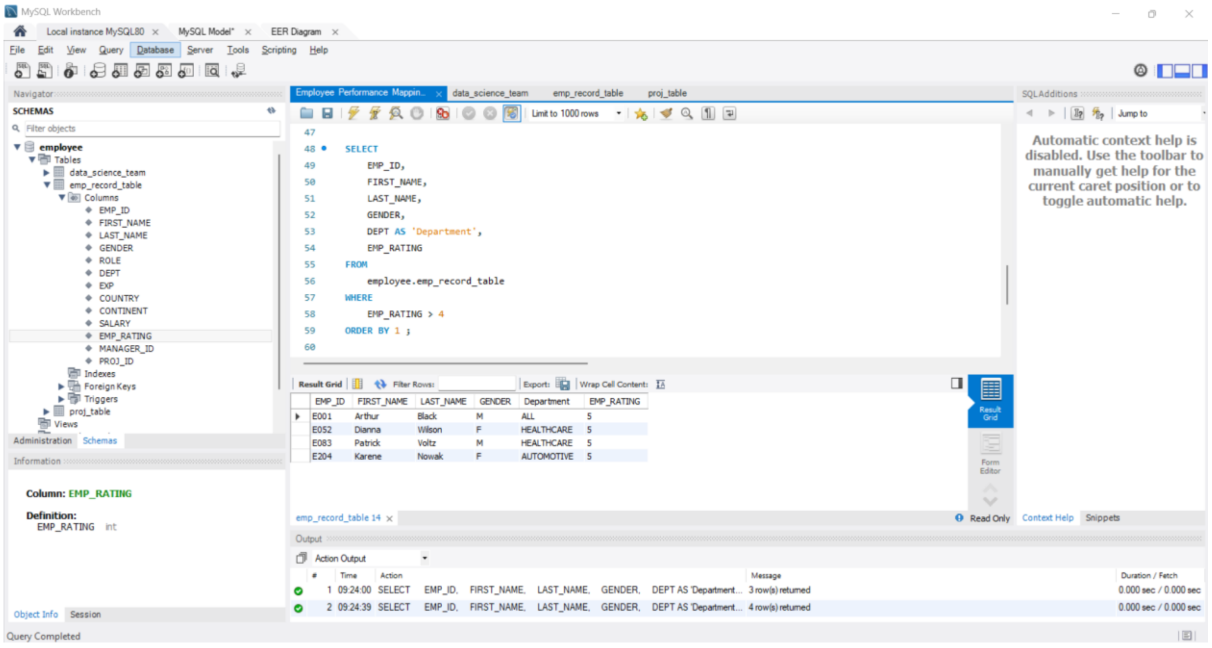

MySQL Workbench
Local instance MySQL80
MySQL Model*
EER Diagram
File Edit View Query Database Server Tools Scripting Help
Employee Performance Mappin...
data_science_team
emp_record_table
proj_table
Limit to 1000 rows
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Column: EMP_RATING
Definition:
EMP_RATING int
Query Completed

```
*/
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    GENDER,
    DEPT AS 'Department',
    EMP_RATING
FROM
    employee.emp_record_table
WHERE
    EMP_RATING < 2
ORDER BY 1 ;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	Department	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  09:24:00  SELECT  EMP_ID,  FIRST_NAME,  LAST_NAME,  GENDER,  DEPT AS 'Department...  3 row(s) returned  0.000 sec / 0.000 sec
```

Greater than four:

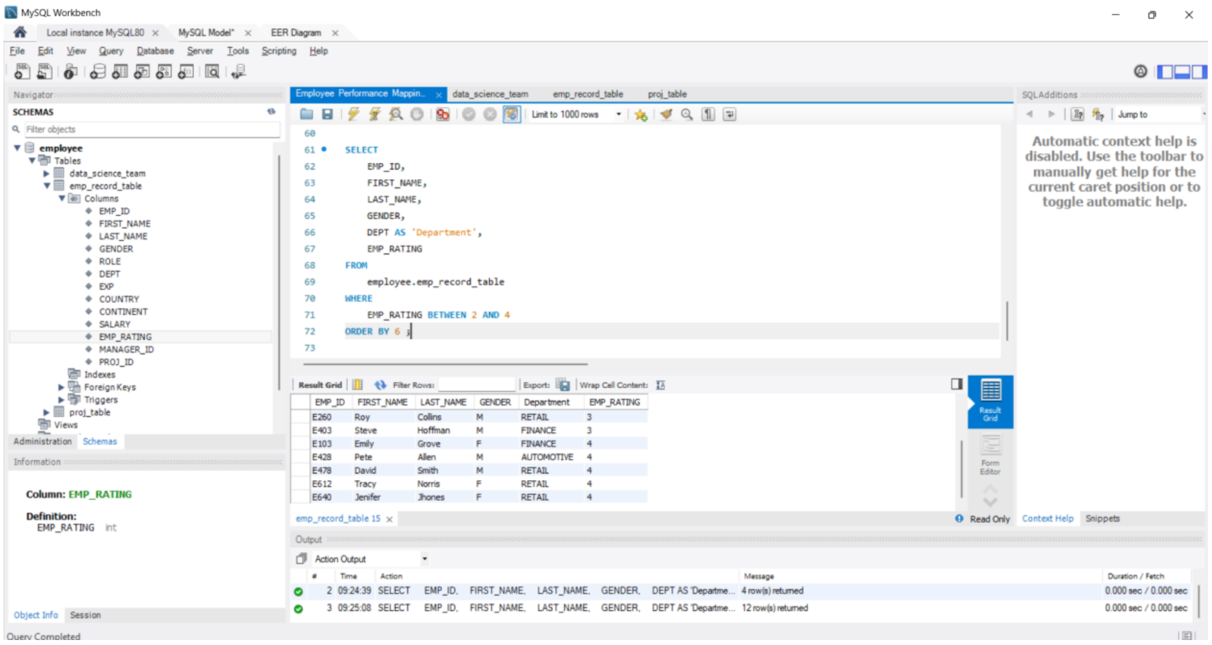

MySQL Workbench
Local instance MySQL80
MySQL Model*
EER Diagram
File Edit View Query Database Server Tools Scripting Help
Employee Performance Mappin...
data_science_team
emp_record_table
proj_table
Limit to 1000 rows
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Column: EMP_RATING
Definition:
EMP_RATING int
Query Completed

```
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    GENDER,
    DEPT AS 'Department',
    EMP_RATING
FROM
    employee.emp_record_table
WHERE
    EMP_RATING > 4
ORDER BY 1 ;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	Department	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  09:24:00  SELECT  EMP_ID,  FIRST_NAME,  LAST_NAME,  GENDER,  DEPT AS 'Department...  3 row(s) returned  0.000 sec / 0.000 sec
2  09:24:39  SELECT  EMP_ID,  FIRST_NAME,  LAST_NAME,  GENDER,  DEPT AS 'Department...  4 row(s) returned  0.000 sec / 0.000 sec
```

Between Two and four:

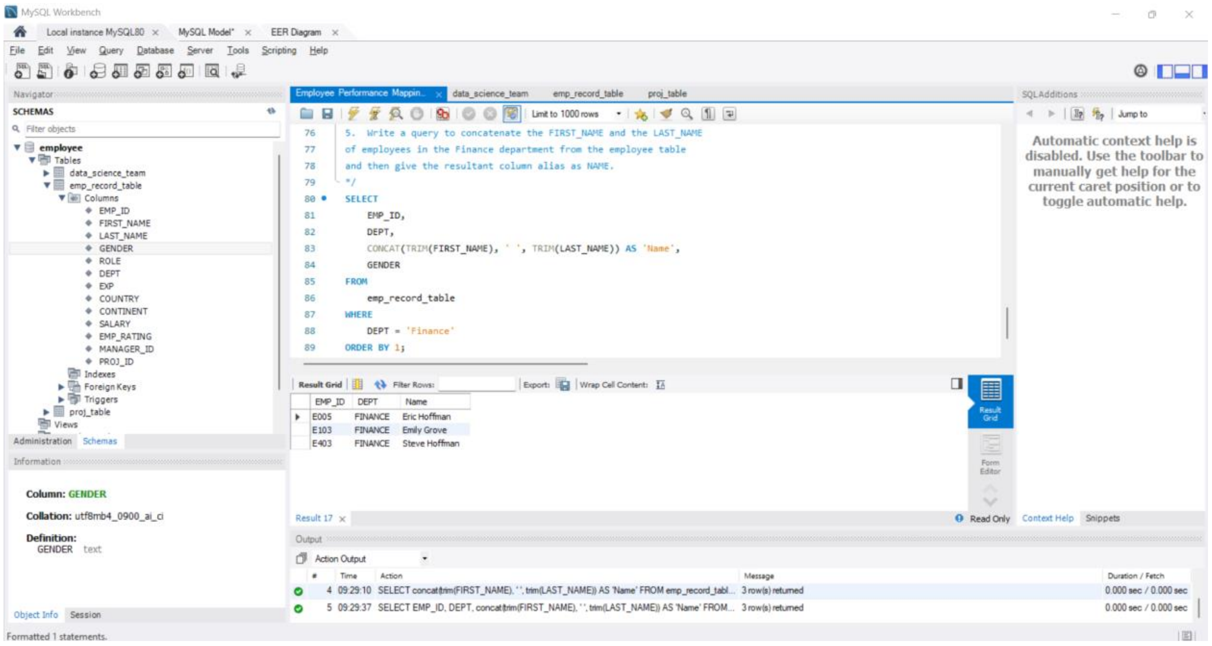


```sql
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    GENDER,
    DEPT AS 'Department',
    EMP_RATING
FROM
    employee.emp_record_table
WHERE
    EMP_RATING BETWEEN 2 AND 4
ORDER BY 6 ;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	Department	EMP_RATING
E260	Roy	Collins	M	RETAIL	3
E403	Steve	Hoffman	M	FINANCE	3
E103	Emily	Grove	F	FINANCE	4
E428	Pete	Allen	M	AUTOMOTIVE	4
E478	David	Smith	M	RETAIL	4
E612	Tracy	Norris	F	RETAIL	4
E640	Jenifer	Jhones	F	RETAIL	4

```
#  Time      Action                                                            Message             Duration / Fetch
2  09:24:39  SELECT  EMP_ID,  FIRST_NAME,  LAST_NAME,  GENDER,  DEPT AS 'Departme...  4 row(s) returned   0.000 sec / 0.000 sec
3  09:25:08  SELECT  EMP_ID,  FIRST_NAME,  LAST_NAME,  GENDER,  DEPT AS 'Departme...  12 row(s) returned  0.000 sec / 0.000 sec
```

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.

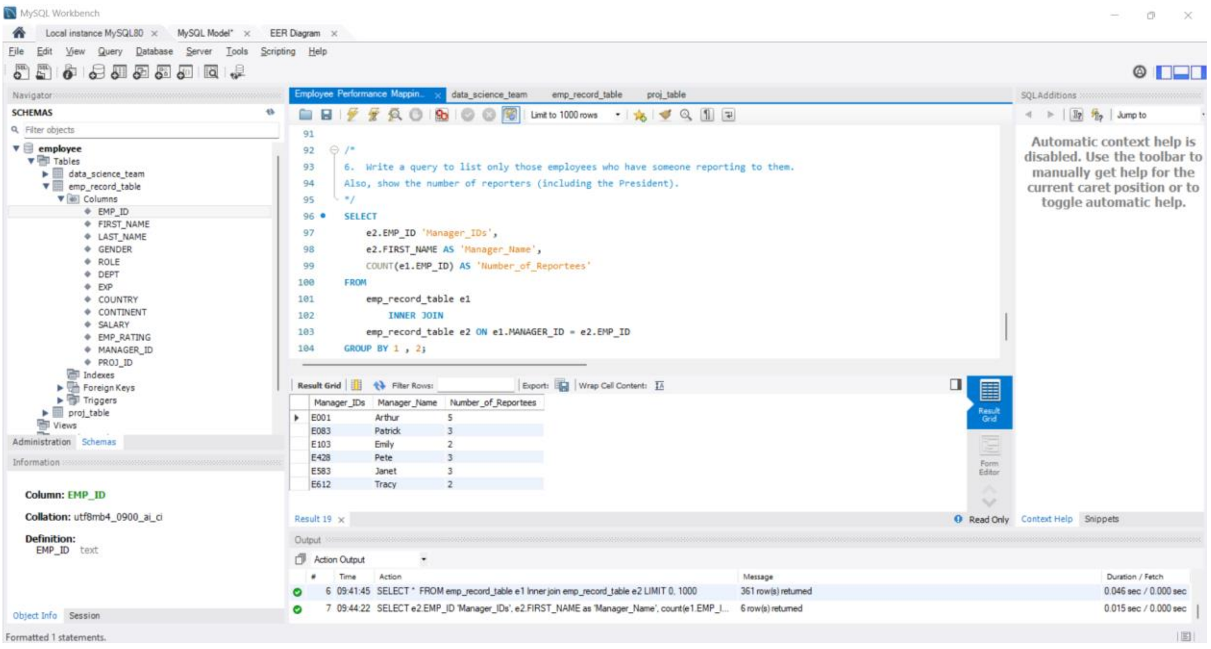


```sql
5.  Write a query to concatenate the FIRST_NAME and the LAST_NAME
of employees in the Finance department from the employee table
and then give the resultant column alias as NAME.
*/
SELECT
    EMP_ID,
    DEPT,
    CONCAT(TRIM(FIRST_NAME), ' ', TRIM(LAST_NAME)) AS 'Name',
    GENDER
FROM
    emp_record_table
WHERE
    DEPT = 'Finance'
ORDER BY 1;
```

EMP_ID	DEPT	Name
E005	FINANCE	Eric Hoffman
E103	FINANCE	Emily Grove
E403	FINANCE	Steve Hoffman

```
#  Time      Action                                                                              Message            Duration / Fetch
4  09:29:10  SELECT concat(trim(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'Name' FROM emp_record_tabl...  3 row(s) returned  0.000 sec / 0.000 sec
5  09:29:37  SELECT EMP_ID, DEPT, concat(trim(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'Name' FROM...    3 row(s) returned  0.000 sec / 0.000 sec
```

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



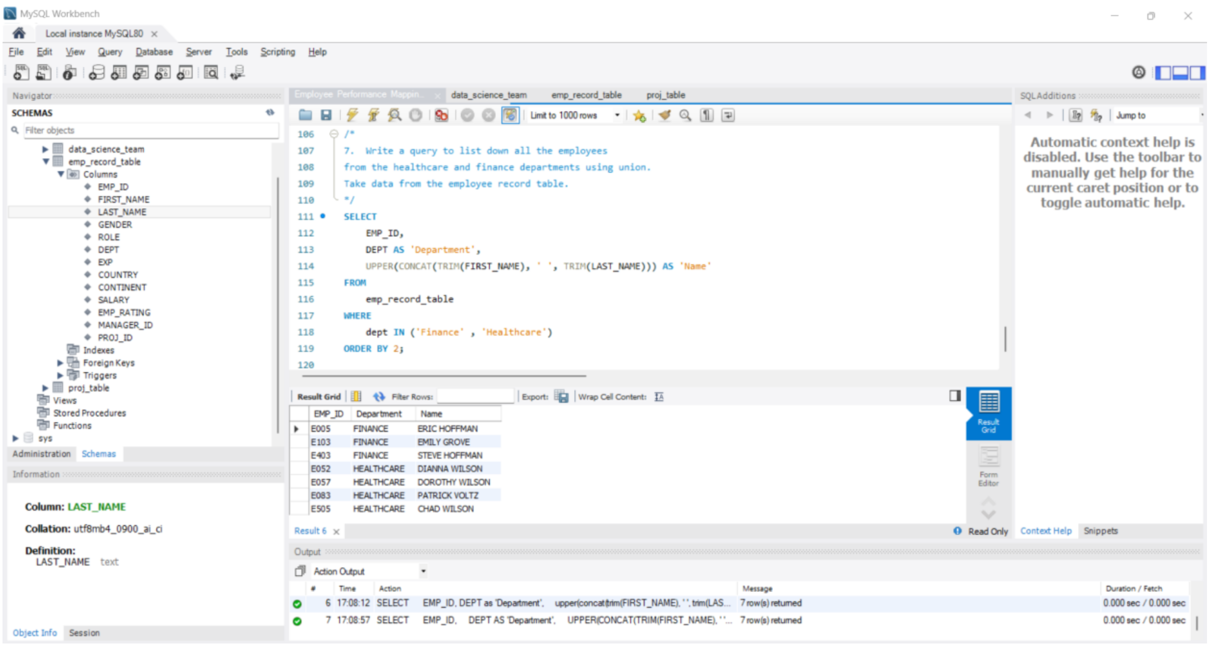
```sql
/*
6.  Write a query to list only those employees who have someone reporting to them.
Also, show the number of reporters (including the President).
*/
SELECT
    e2.EMP_ID 'Manager_IDs',
    e2.FIRST_NAME AS 'Manager_Name',
    COUNT(e1.EMP_ID) AS 'Number_of_Reportees'
FROM
    emp_record_table e1
        INNER JOIN
    emp_record_table e2 ON e1.MANAGER_ID = e2.EMP_ID
GROUP BY 1 , 2;
```

Manager_IDs	Manager_Name	Number_of_Reportees
E001	Arthur	5
E083	Patrick	3
E103	Emily	2
E428	Pete	3
E583	Janet	3
E612	Tracy	2

```
#  Time      Action                                                                  Message              Duration / Fetch
6  09:41:45  SELECT * FROM emp_record_table e1 Inner join emp_record_table e2 LIMIT 0, 1000   361 row(s) returned  0.046 sec / 0.000 sec
7  09:44:22  SELECT e2.EMP_ID 'Manager_IDs', e2.FIRST_NAME as 'Manager_Name', count(e1.EMP_I...   6 row(s) returned    0.015 sec / 0.000 sec
```

Formatted 1 statements.

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.

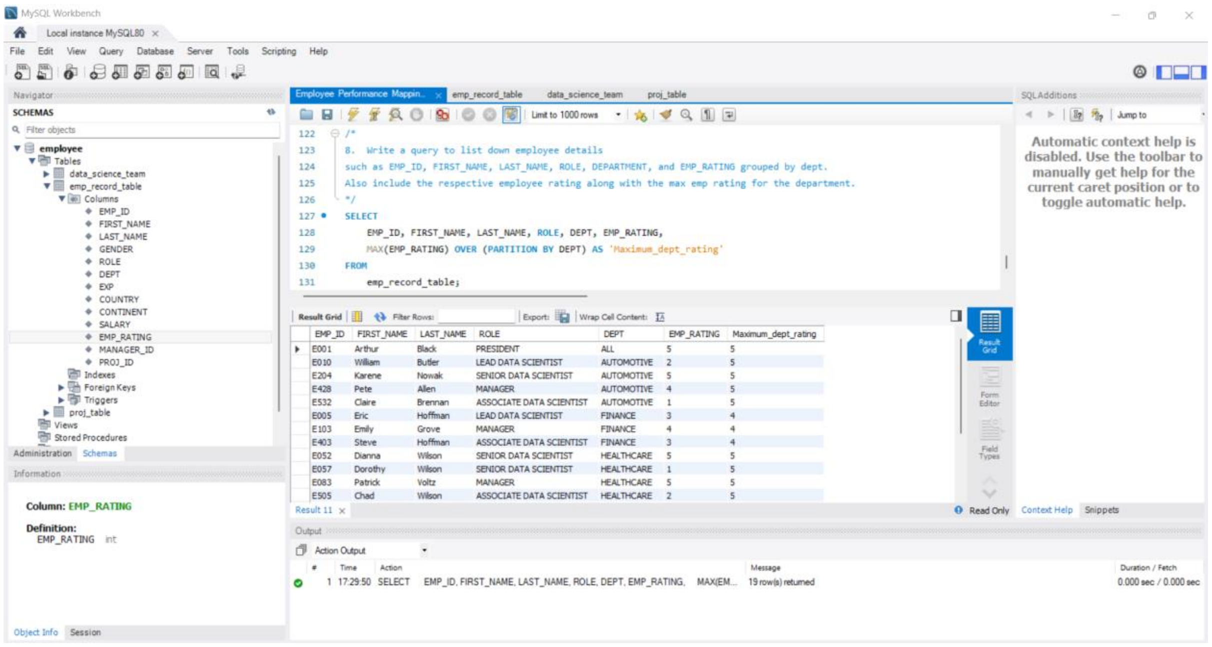


```sql
/*
7.  Write a query to list down all the employees
from the healthcare and finance departments using union.
Take data from the employee record table.
*/
SELECT
    EMP_ID,
    DEPT AS 'Department',
    UPPER(CONCAT(TRIM(FIRST_NAME), ' ', TRIM(LAST_NAME))) AS 'Name'
FROM
    emp_record_table
WHERE
    dept IN ('Finance' , 'Healthcare')
ORDER BY 2;
```

EMP_ID	Department	Name
E005	FINANCE	ERIC HOFFMAN
E103	FINANCE	EMILY GROVE
E403	FINANCE	STEVE HOFFMAN
E052	HEALTHCARE	DIANNA WILSON
E057	HEALTHCARE	DOROTHY WILSON
E083	HEALTHCARE	PATRICK VOLTZ
E505	HEALTHCARE	CHAD WILSON

```
#  Time      Action                                                                        Message            Duration / Fetch
6  17:08:12  SELECT    EMP_ID, DEPT as 'Department',    upper(concat(trim(FIRST_NAME), ' ', trim(LAS...   7 row(s) returned  0.000 sec / 0.000 sec
7  17:08:57  SELECT    EMP_ID,    DEPT AS 'Department',    UPPER(CONCAT(TRIM(FIRST_NAME), ' '...   7 row(s) returned  0.000 sec / 0.000 sec
```

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.



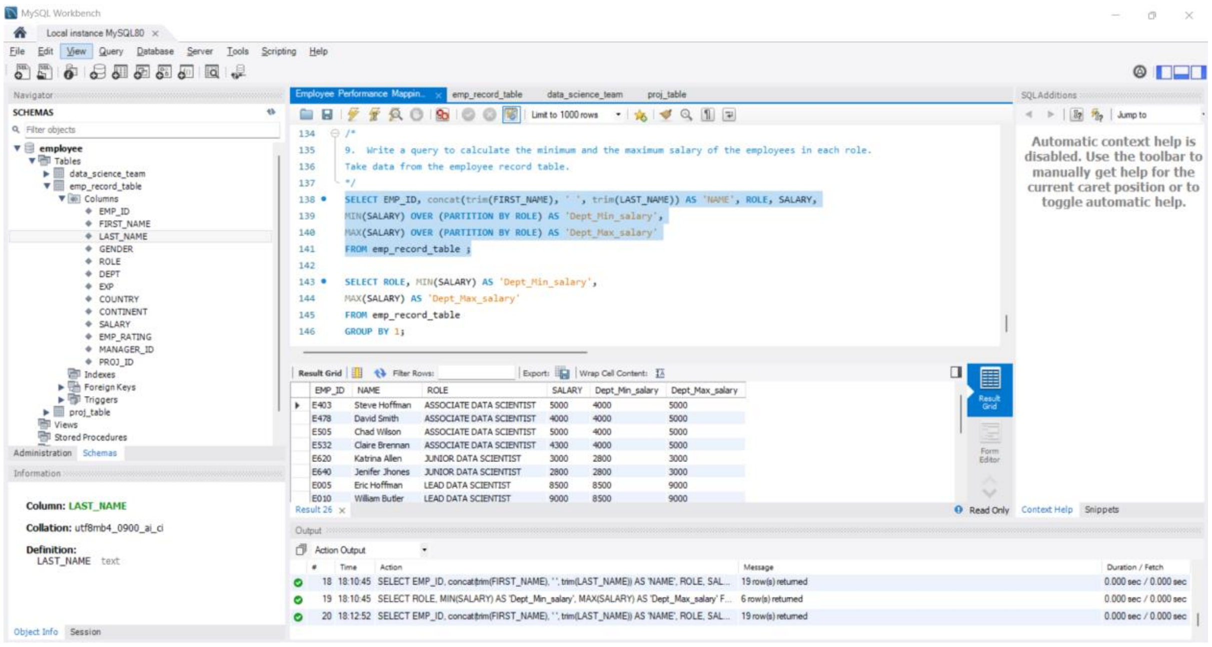
```
/*
8.  Write a query to list down employee details
such as EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING grouped by dept.
Also include the respective employee rating along with the max emp rating for the department.
*/
SELECT
    EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING,
    MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS 'Maximum_dept_rating'
FROM
    emp_record_table;
```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	Maximum_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

```
#  Time      Action                                                           Message
1  17:29:50  SELECT  EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING,  MAX(EM...  19 row(s) returned   0.000 sec / 0.000 sec
```

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.

Option 1: with Window Function (Preferred if we need employee-level details)

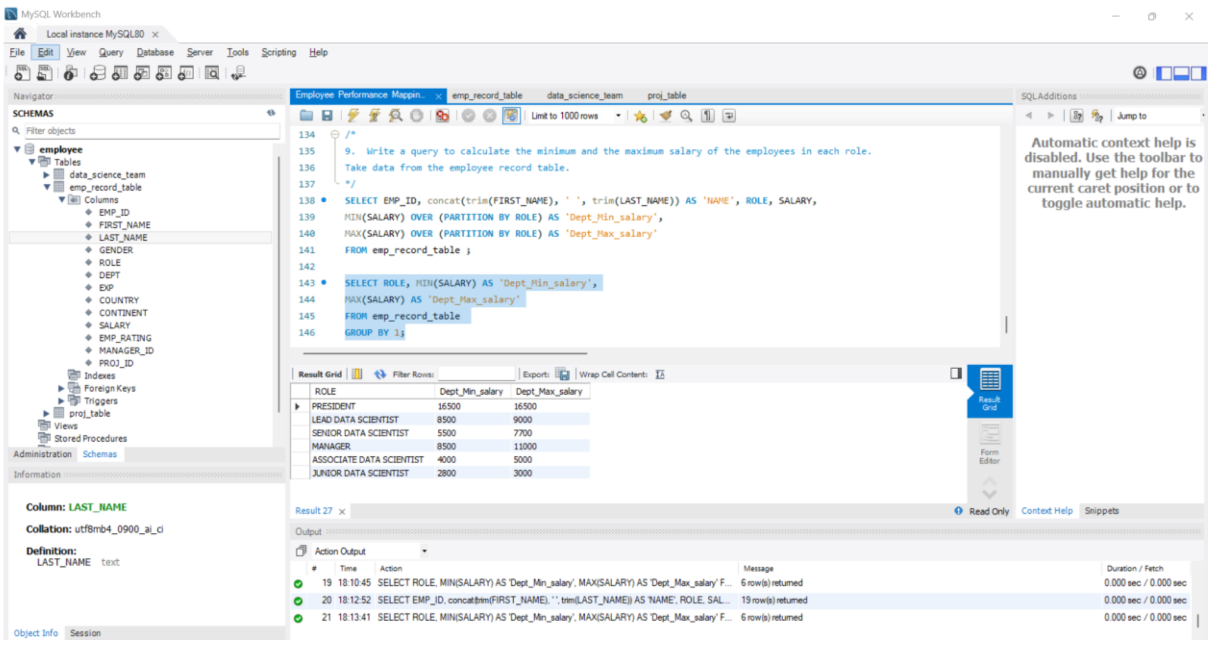


```
/*
9.  Write a query to calculate the minimum and the maximum salary of the employees in each role.
Take data from the employee record table.
*/
SELECT EMP_ID, concat(trim(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'NAME', ROLE, SALARY,
MIN(SALARY) OVER (PARTITION BY ROLE) AS 'Dept_Min_salary',
MAX(SALARY) OVER (PARTITION BY ROLE) AS 'Dept_Max_salary'
FROM emp_record_table ;

SELECT ROLE, MIN(SALARY) AS 'Dept_Min_salary',
MAX(SALARY) AS 'Dept_Max_salary'
FROM emp_record_table
GROUP BY 1;
```

EMP_ID	NAME	ROLE	SALARY	Dept_Min_salary	Dept_Max_salary
E403	Steve Hoffman	ASSOCIATE DATA SCIENTIST	5000	4000	5000
E478	David Smith	ASSOCIATE DATA SCIENTIST	4000	4000	5000
E505	Chad Wilson	ASSOCIATE DATA SCIENTIST	5000	4000	5000
E532	Claire Brennan	ASSOCIATE DATA SCIENTIST	4300	4000	5000
E620	Katrina Allen	JUNIOR DATA SCIENTIST	3000	2800	3000
E640	Jenifer Jhones	JUNIOR DATA SCIENTIST	2800	2800	3000
E005	Eric Hoffman	LEAD DATA SCIENTIST	8500	8500	9000
E010	William Butler	LEAD DATA SCIENTIST	9000	8500	9000

```
#   Time      Action                                                                        Message              Duration / Fetch
18  18:10:45  SELECT EMP_ID, concat(trim(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'NAME', ROLE, SAL...  19 row(s) returned   0.000 sec / 0.000 sec
19  18:10:45  SELECT ROLE, MIN(SALARY) AS 'Dept_Min_salary', MAX(SALARY) AS 'Dept_Max_salary' F...  6 row(s) returned    0.000 sec / 0.000 sec
20  18:12:52  SELECT EMP_ID, concat(trim(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'NAME', ROLE, SAL...  19 row(s) returned   0.000 sec / 0.000 sec
```

Option 2: With Group by (Cleaner, and precise as per the question)

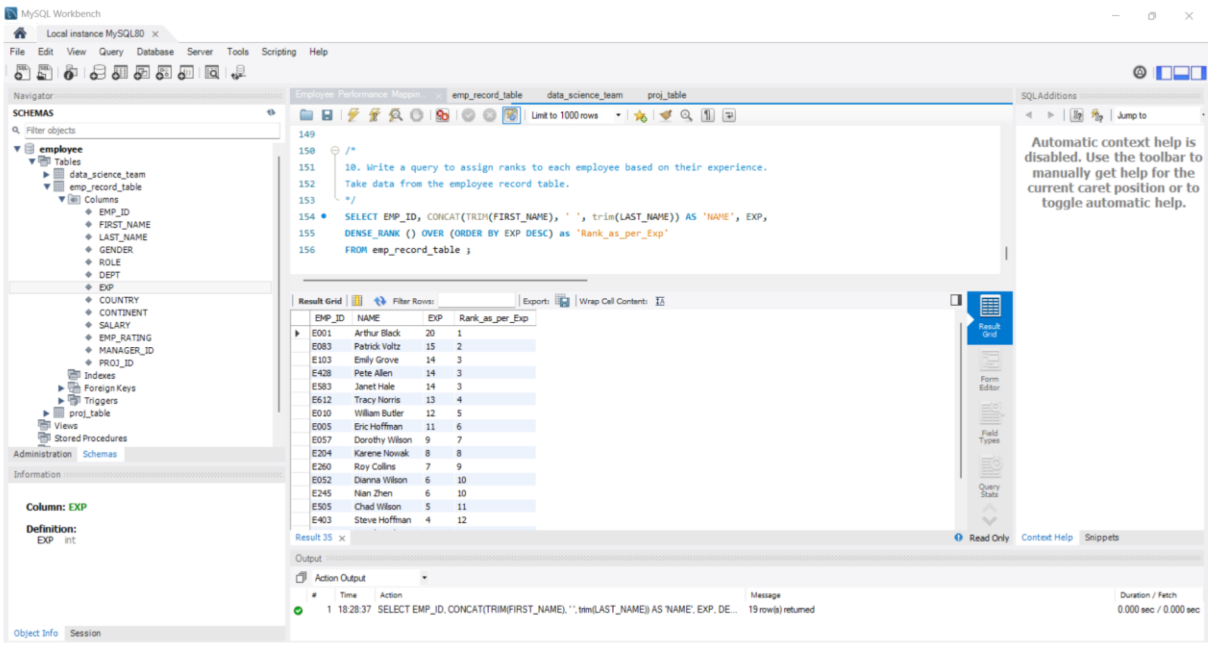


```
/*
9.  Write a query to calculate the minimum and the maximum salary of the employees in each role.
Take data from the employee record table.
*/
SELECT EMP_ID, concat(trim(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'NAME', ROLE, SALARY,
MIN(SALARY) OVER (PARTITION BY ROLE) AS 'Dept_Min_salary',
MAX(SALARY) OVER (PARTITION BY ROLE) AS 'Dept_Max_salary'
FROM emp_record_table ;

SELECT ROLE, MIN(SALARY) AS 'Dept_Min_salary',
MAX(SALARY) AS 'Dept_Max_salary'
FROM emp_record_table
GROUP BY 1;
```

ROLE	Dept_Min_salary	Dept_Max_salary
PRESIDENT	16500	16500
LEAD DATA SCIENTIST	8500	9000
SENIOR DATA SCIENTIST	5500	7700
MANAGER	8500	11000
ASSOCIATE DATA SCIENTIST	4000	5000
JUNIOR DATA SCIENTIST	2800	3000

```
#   Time      Action                                                                        Message             Duration / Fetch
19  18:10:45  SELECT ROLE, MIN(SALARY) AS 'Dept_Min_salary', MAX(SALARY) AS 'Dept_Max_salary' F...  6 row(s) returned   0.000 sec / 0.000 sec
20  18:12:52  SELECT EMP_ID, concat(trim(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'NAME', ROLE, SAL...  19 row(s) returned  0.000 sec / 0.000 sec
21  18:13:41  SELECT ROLE, MIN(SALARY) AS 'Dept_Min_salary', MAX(SALARY) AS 'Dept_Max_salary' F...  6 row(s) returned   0.000 sec / 0.000 sec
```

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

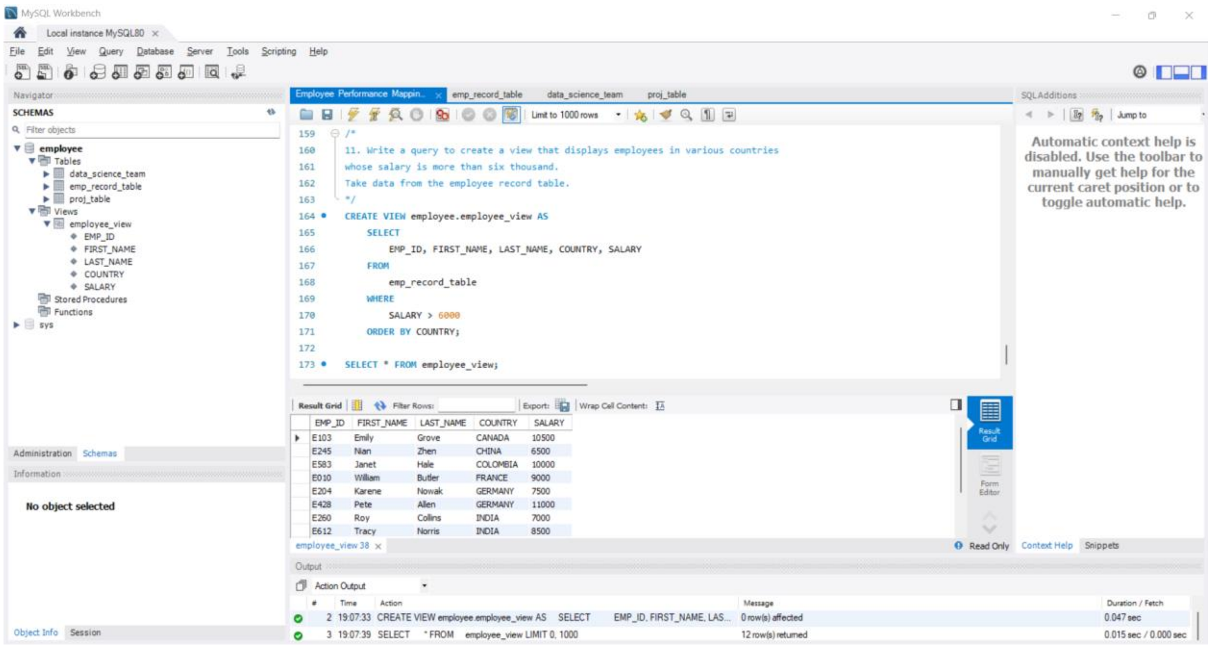


```
/*
10. Write a query to assign ranks to each employee based on their experience.
Take data from the employee record table.
*/
SELECT EMP_ID, CONCAT(TRIM(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'NAME', EXP,
DENSE_RANK () OVER (ORDER BY EXP DESC) as 'Rank_as_per_Exp'
FROM emp_record_table ;
```

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

```
#  Time      Action                                                                     Message             Duration / Fetch
1  18:28:37  SELECT EMP_ID, CONCAT(TRIM(FIRST_NAME), ' ', trim(LAST_NAME)) AS 'NAME', EXP, DE...  19 row(s) returned  0.000 sec / 0.000 sec
```

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.

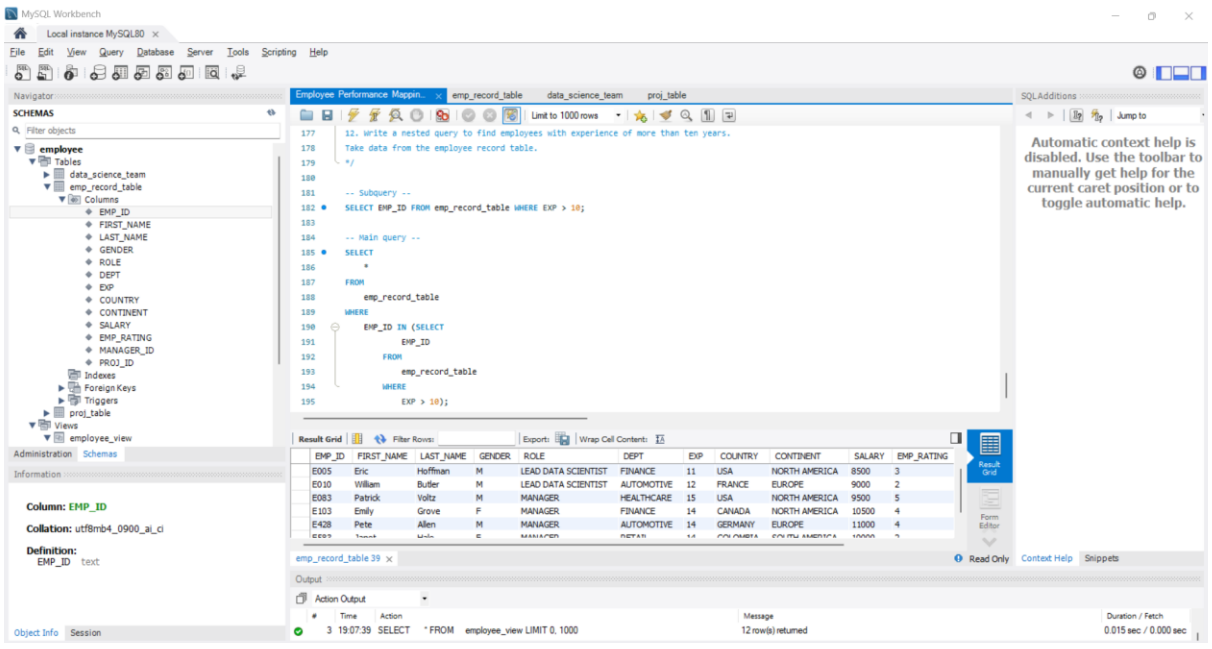


```
/*
11. Write a query to create a view that displays employees in various countries
whose salary is more than six thousand.
Take data from the employee record table.
*/
CREATE VIEW employee.employee_view AS
    SELECT
        EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY
    FROM
        emp_record_table
    WHERE
        SALARY > 6000
    ORDER BY COUNTRY;

SELECT * FROM employee_view;
```

EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
E103	Emily	Grove	CANADA	10500
E245	Nian	Zhen	CHINA	6500
E583	Janet	Hale	COLOMBIA	10000
E010	William	Butler	FRANCE	9000
E204	Karene	Nowak	GERMANY	7500
E428	Pete	Allen	GERMANY	11000
E260	Roy	Collins	INDIA	7000
E612	Tracy	Norris	INDIA	8500

```
#  Time      Action                                                    Message             Duration / Fetch
2  19:07:33  CREATE VIEW employee.employee_view AS  SELECT  EMP_ID, FIRST_NAME, LAS...  0 row(s) affected  0.047 sec
3  19:07:39  SELECT  * FROM  employee_view LIMIT 0, 1000                12 row(s) returned  0.015 sec / 0.000 sec
```

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

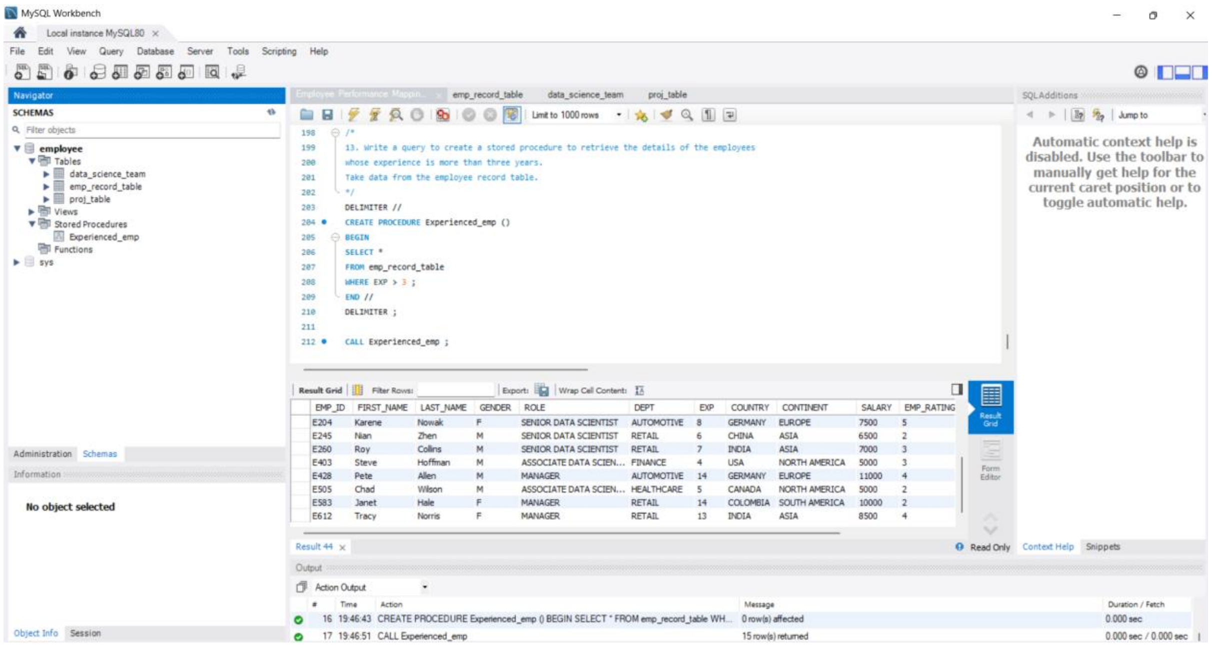


```
12. Write a nested query to find employees with experience of more than ten years.
Take data from the employee record table.
*/

-- Subquery --
SELECT EMP_ID FROM emp_record_table WHERE EXP > 10;

-- Main query --
SELECT
    *
FROM
    emp_record_table
WHERE
    EMP_ID IN (SELECT
            EMP_ID
        FROM
            emp_record_table
        WHERE
            EXP > 10);
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4
E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4

```
#  Time      Action                                       Message             Duration / Fetch
3  19:07:39  SELECT  * FROM  employee_view LIMIT 0, 1000  12 row(s) returned  0.015 sec / 0.000 sec
```

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.



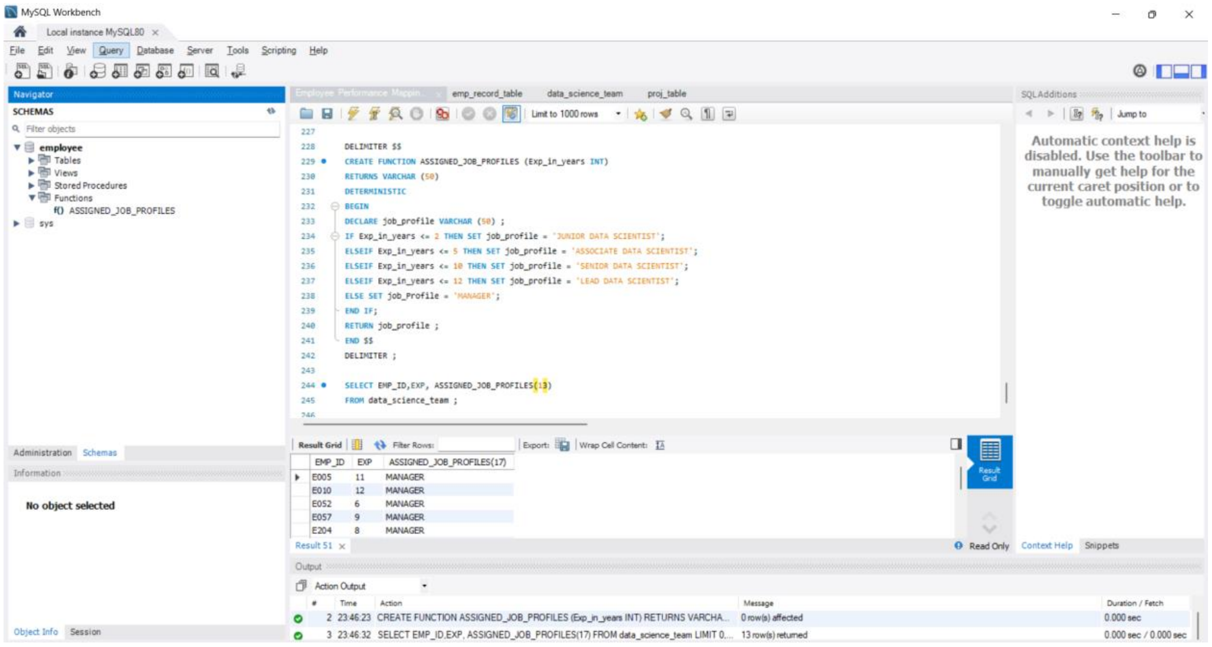
```
/*
13. Write a query to create a stored procedure to retrieve the details of the employees
whose experience is more than three years.
Take data from the employee record table.
*/
DELIMITER //
CREATE PROCEDURE Experienced_emp ()
BEGIN
SELECT *
FROM emp_record_table
WHERE EXP > 3 ;
END //
DELIMITER ;

CALL Experienced_emp ;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING
E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5
E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500	2
E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIEN...	FINANCE	4	USA	NORTH AMERICA	5000	3
E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4
E505	Chad	Wilson	M	ASSOCIATE DATA SCIEN...	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2
E583	Janet	Hale	F	MANAGER	RETAIL	14	COLOMBIA	SOUTH AMERICA	10000	2
E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4

```
#   Time      Action                                                                  Message              Duration / Fetch
16  19:46:43  CREATE PROCEDURE Experienced_emp () BEGIN SELECT * FROM emp_record_table WH...  0 row(s) affected    0.000 sec
17  19:46:51  CALL Experienced_emp                                                    15 row(s) returned   0.000 sec / 0.000 sec
```

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'.



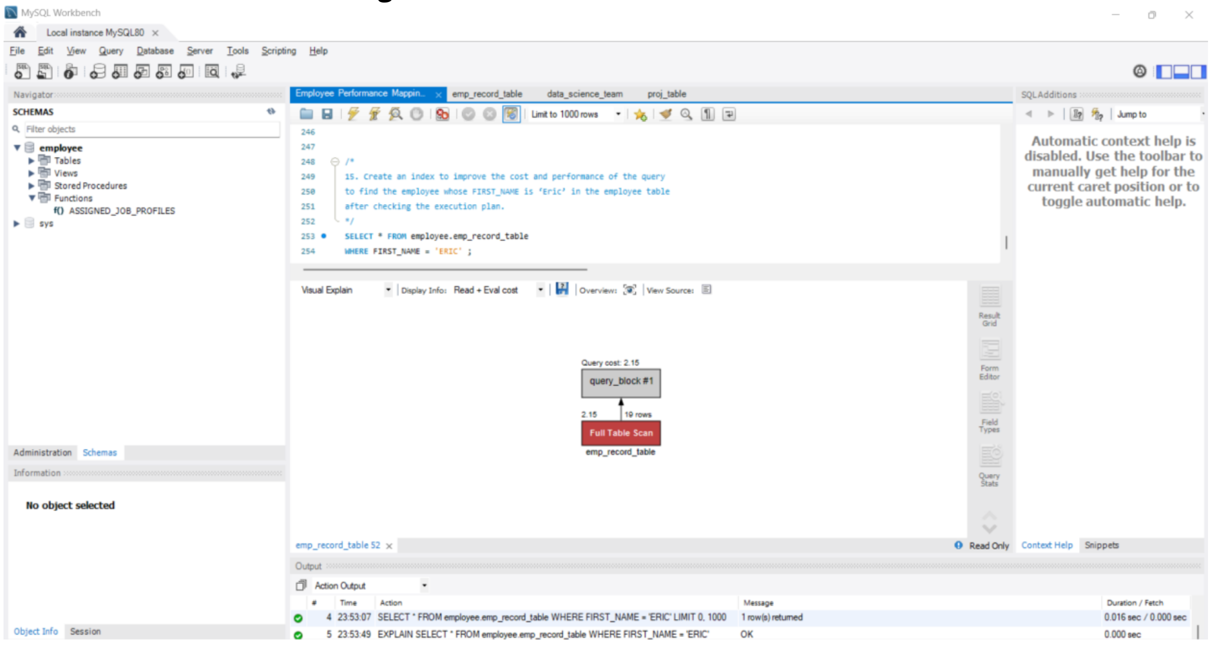
```
DELIMITER $$
CREATE FUNCTION ASSIGNED_JOB_PROFILES (Exp_in_years INT)
RETURNS VARCHAR (50)
DETERMINISTIC
BEGIN
DECLARE job_profile VARCHAR (50) ;
IF Exp_in_years <= 2 THEN SET job_profile = 'JUNIOR DATA SCIENTIST';
ELSEIF Exp_in_years <= 5 THEN SET job_profile = 'ASSOCIATE DATA SCIENTIST';
ELSEIF Exp_in_years <= 10 THEN SET job_profile = 'SENIOR DATA SCIENTIST';
ELSEIF Exp_in_years <= 12 THEN SET job_profile = 'LEAD DATA SCIENTIST';
ELSE SET job_Profile = 'MANAGER';
END IF;
RETURN job_profile ;
END $$
DELIMITER ;

SELECT EMP_ID,EXP, ASSIGNED_JOB_PROFILES(13)
FROM data_science_team ;
```

EMP_ID	EXP	ASSIGNED_JOB_PROFILES(17)
E005	11	MANAGER
E010	12	MANAGER
E052	6	MANAGER
E057	9	MANAGER
E204	8	MANAGER

```
#  Time      Action                                                                   Message              Duration / Fetch
2  23:46:23  CREATE FUNCTION ASSIGNED_JOB_PROFILES (Exp_in_years INT) RETURNS VARCHA...  0 row(s) affected    0.000 sec
3  23:46:32  SELECT EMP_ID,EXP, ASSIGNED_JOB_PROFILES(17) FROM data_science_team LIMIT 0,...  13 row(s) returned   0.000 sec / 0.000 sec
```

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.

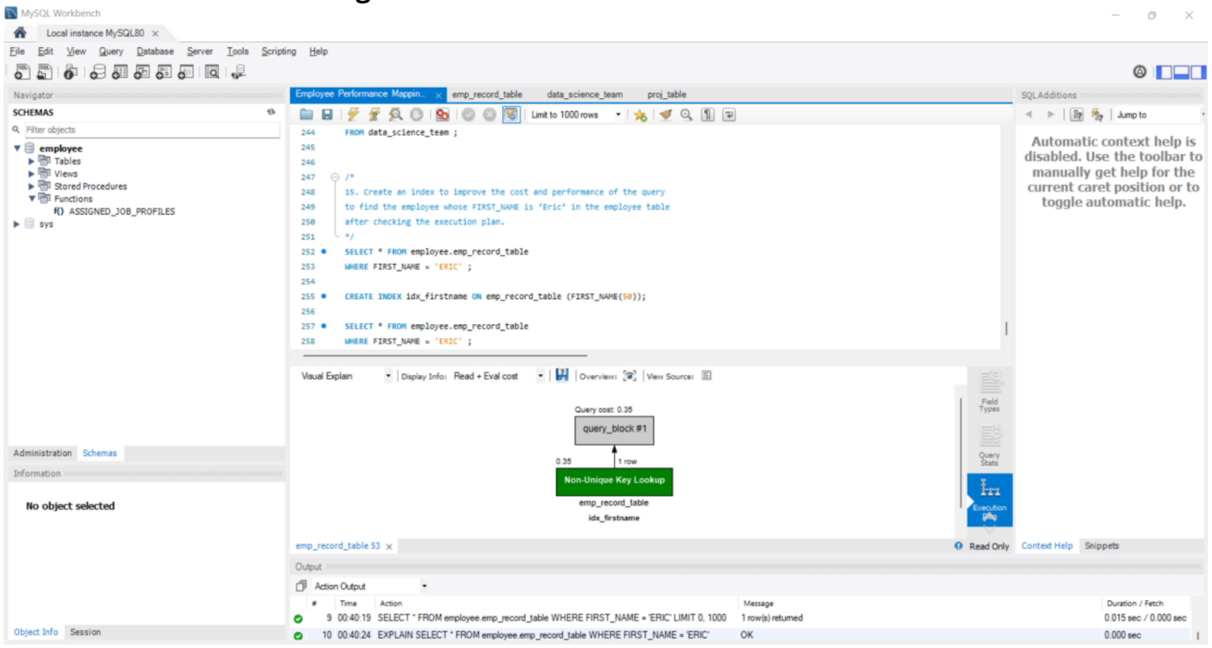
Performance before Indexing:



```
/*
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.
*/
SELECT * FROM employee.emp_record_table
WHERE FIRST_NAME = 'ERIC' ;
```

```
#  Time      Action                                                                        Message              Duration / Fetch
4  23:53:07  SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = 'ERIC' LIMIT 0, 1000  1 row(s) returned    0.016 sec / 0.000 sec
5  23:53:49  EXPLAIN SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = 'ERIC'         OK                   0.000 sec
```

Performance after Indexing:



```
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.
*/
SELECT * FROM employee.emp_record_table
WHERE FIRST_NAME = 'ERIC' ;

CREATE INDEX idx_firstname ON emp_record_table (FIRST_NAME(50));

SELECT * FROM employee.emp_record_table
WHERE FIRST_NAME = 'ERIC' ;
```

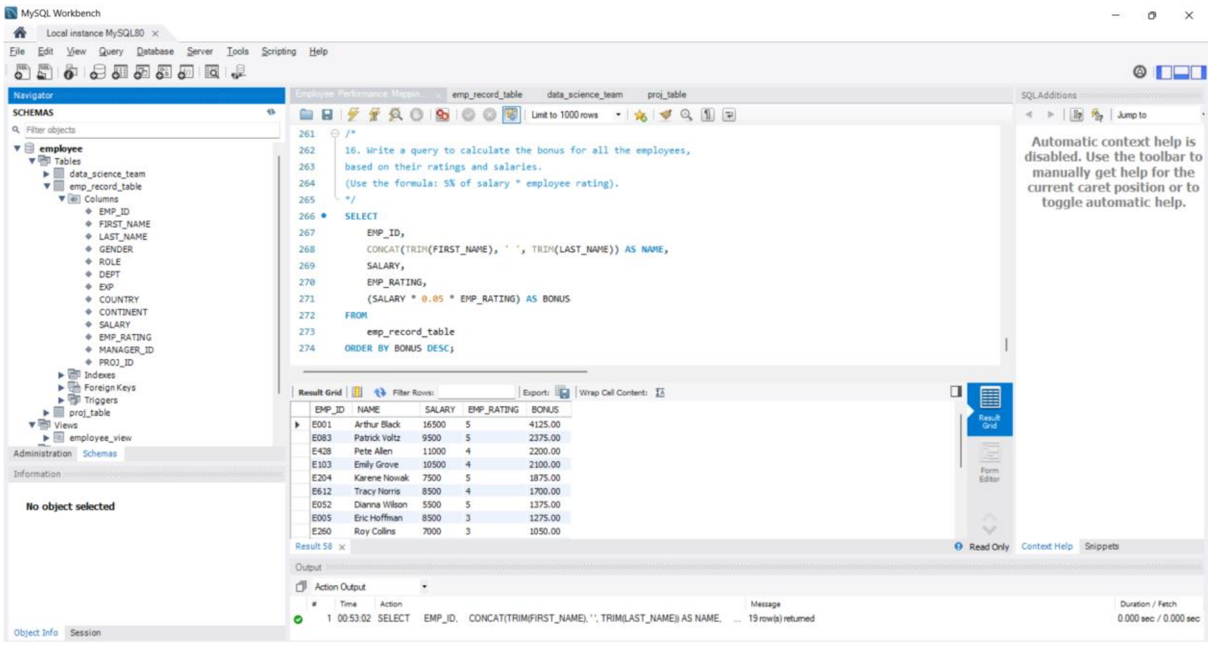
```
#   Time      Action                                                                        Message              Duration / Fetch
9   00:40:19  SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = 'ERIC' LIMIT 0, 1000  1 row(s) returned    0.015 sec / 0.000 sec
10  00:40:24  EXPLAIN SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = 'ERIC'         OK                   0.000 sec
```

Observation:

Before indexing, the query to find employees with first name *Eric* scanned the entire table, resulting in higher execution cost (2.15).

After creating an index on the **FIRST_NAME** column, the execution plan showed that the query used the index, reducing the query execution time and cost significantly (0.35).

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).

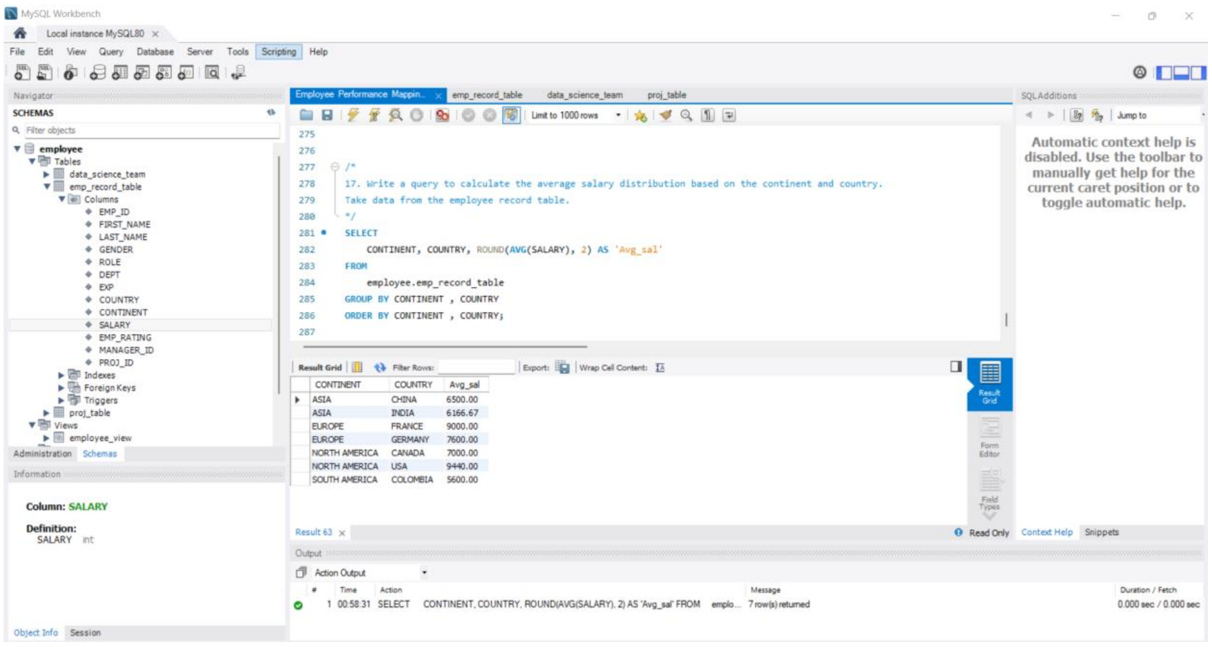


```
/*
16. Write a query to calculate the bonus for all the employees,
based on their ratings and salaries.
(Use the formula: 5% of salary * employee rating).
*/
SELECT
    EMP_ID,
    CONCAT(TRIM(FIRST_NAME), ' ', TRIM(LAST_NAME)) AS NAME,
    SALARY,
    EMP_RATING,
    (SALARY * 0.05 * EMP_RATING) AS BONUS
FROM
    emp_record_table
ORDER BY BONUS DESC;
```

EMP_ID	NAME	SALARY	EMP_RATING	BONUS
E001	Arthur Black	16500	5	4125.00
E083	Patrick Voltz	9500	5	2375.00
E428	Pete Allen	11000	4	2200.00
E103	Emily Grove	10500	4	2100.00
E204	Karene Nowak	7500	5	1875.00
E612	Tracy Norris	8500	4	1700.00
E052	Dianna Wilson	5500	5	1375.00
E005	Eric Hoffman	8500	3	1275.00
E260	Roy Collins	7000	3	1050.00

```
1  00:53:02  SELECT  EMP_ID,  CONCAT(TRIM(FIRST_NAME), ' ', TRIM(LAST_NAME)) AS NAME,  ...  19 row(s) returned  0.000 sec / 0.000 sec
```

17. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee 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.
*/
SELECT
    CONTINENT, COUNTRY, ROUND(AVG(SALARY), 2) AS 'Avg_sal'
FROM
    employee.emp_record_table
GROUP BY CONTINENT , COUNTRY
ORDER BY CONTINENT , COUNTRY;
```

CONTINENT	COUNTRY	Avg_sal
ASIA	CHINA	6500.00
ASIA	INDIA	6166.67
EUROPE	FRANCE	9000.00
EUROPE	GERMANY	7600.00
NORTH AMERICA	CANADA	7000.00
NORTH AMERICA	USA	9440.00
SOUTH AMERICA	COLOMBIA	5600.00

```
1  00:58:31  SELECT  CONTINENT, COUNTRY, ROUND(AVG(SALARY), 2) AS 'Avg_sal' FROM  emplo...  7 row(s) returned  0.000 sec / 0.000 sec
```

Conclusion:

In this project, the **Employee database** was created and analysed through various SQL queries. The tasks included:

- Designing an ER diagram to represent the database structure.
- Writing queries to fetch and filter employee data.
- Applying aggregation, grouping, and ranking functions.
- Creating views, nested queries, and stored procedures for reusability.
- Implementing stored functions to validate job profiles against company standards.
- Using indexing to improve query performance.
- Calculating salary-based bonuses and analysing salary distributions by country and continent.

Through these implementations, the project demonstrates practical applications of SQL in performance mapping and database optimization.