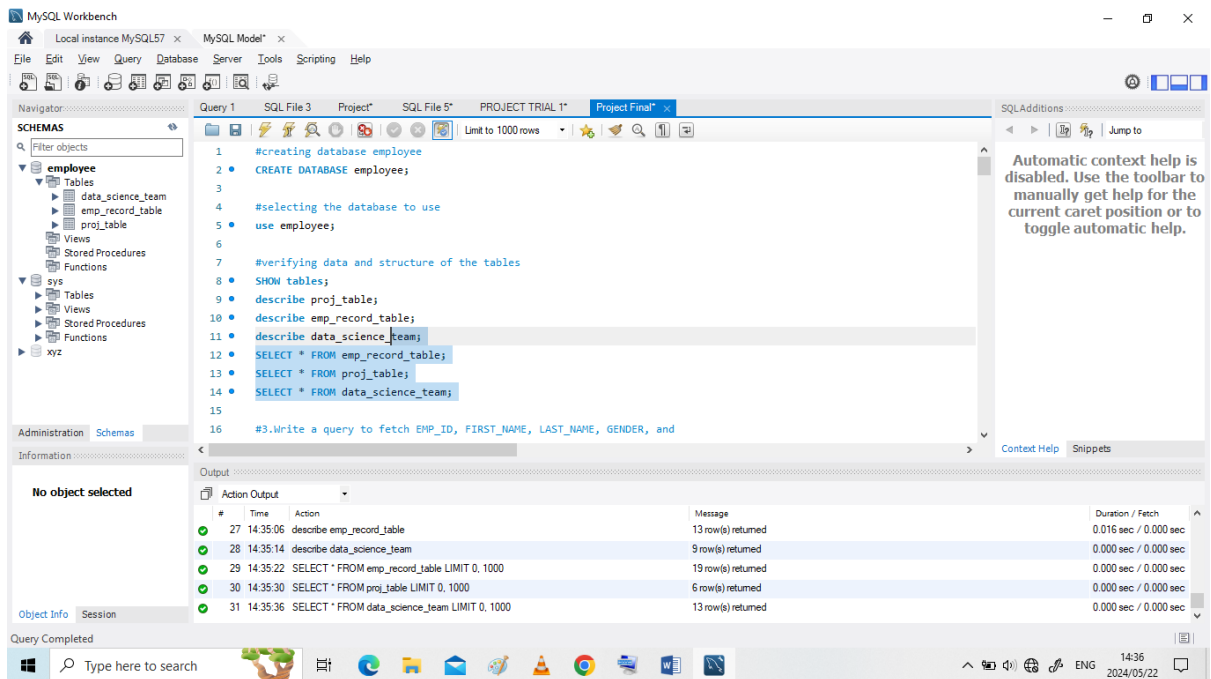
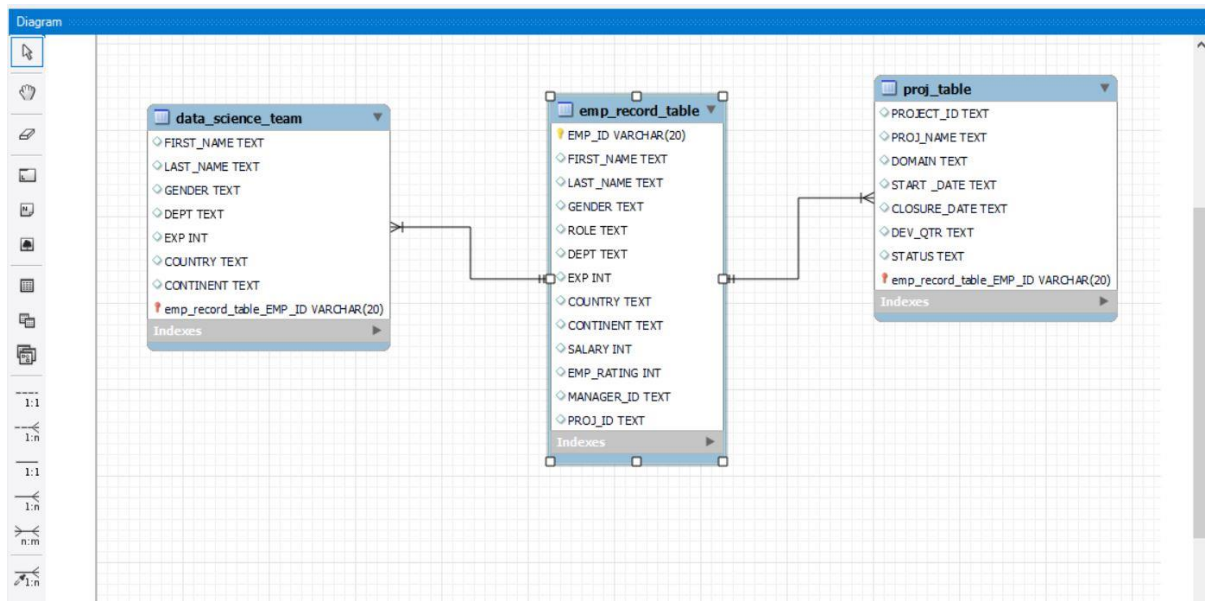


1. Create a database named **employee**, then import **data_science_team.csv**, **proj_table.csv** and **emp_record_table.csv** into the **employee** database from the given resources.

CREATE DATABASE **employee**;

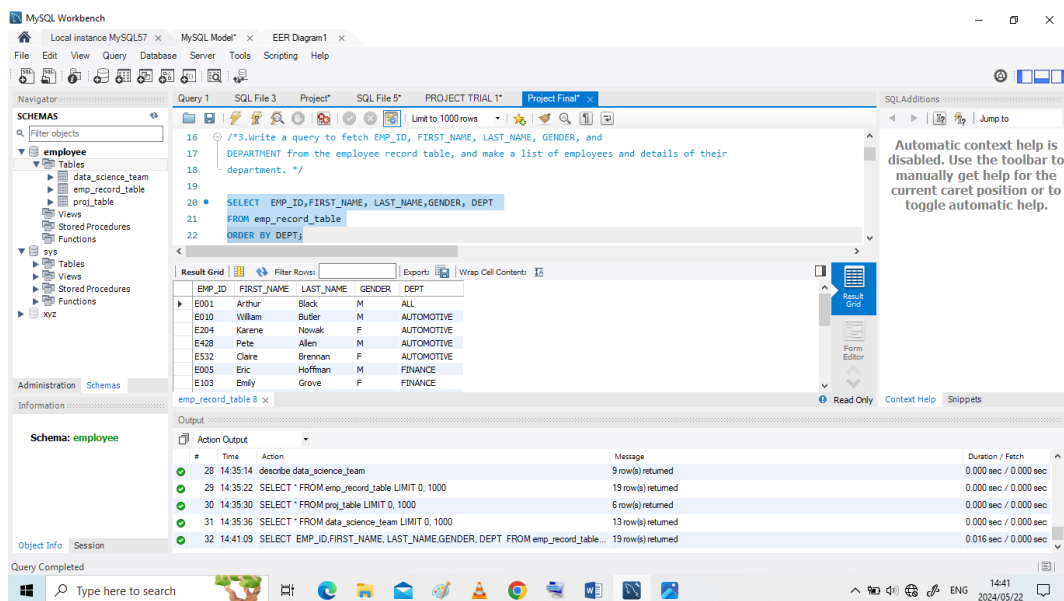


2. Create an ER diagram for the given **employee** database.



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.

```
SELECT EMP_ID,FIRST_NAME, LAST_NAME,GENDER, DEPT  
  
FROM emp_record_table  
  
ORDER BY DEPT;
```



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

#If rating is less than 2

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

#If rating is greater than 4

```
SELECT EMP_ID,FIRST_NAME, LAST_NAME,GENDER, DEPT,EMP_RATING  
  
FROM emp_record_table  
  
WHERE EMP_RATING >4;
```

#Between 2 and 4

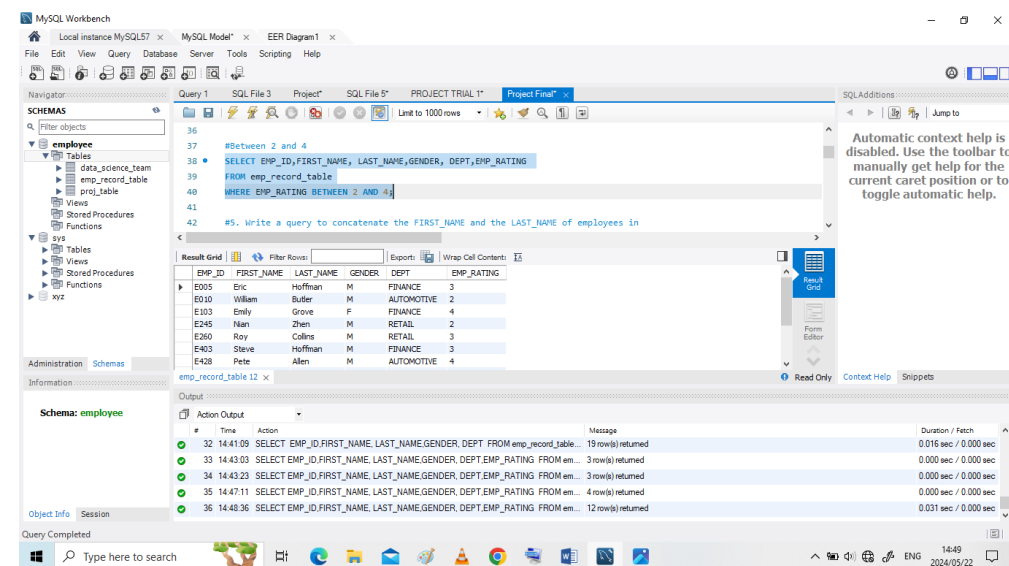
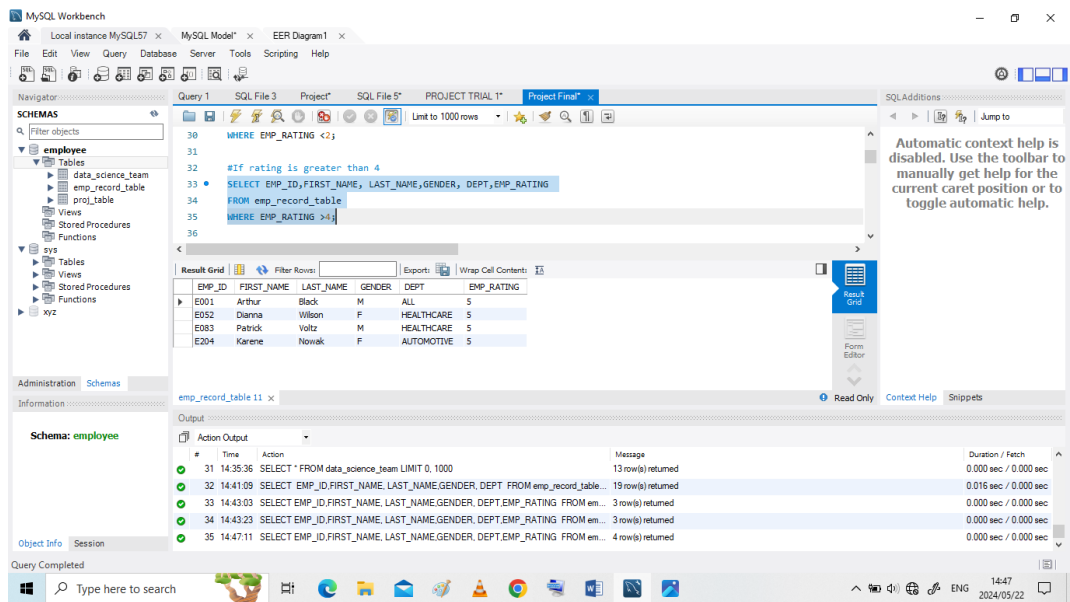
```
SELECT EMP_ID,FIRST_NAME, LAST_NAME,GENDER, DEPT,EMP_RATING  
  
FROM emp_record_table  
  
WHERE EMP_RATING BETWEEN 2 AND 4;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains a query with a comment and a WHERE clause: `SELECT EMP_ID,FIRST_NAME, LAST_NAME,GENDER, DEPT,EMP_RATING FROM emp_record_table WHERE EMP_RATING <2;`. The Results Grid displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
ES17	Dorothy	Wilson	F	HEALTHCARE	1
ES32	Claire	Brennan	F	AUTOMOTIVE	1
ES20	Katrina	Allen	F	RETAIL	1

The Output panel at the bottom shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
29	14:35:22	SELECT * FROM emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
30	14:35:30	SELECT * FROM proj_table LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
31	14:35:36	SELECT * FROM data_science_team LIMIT 0, 1000	13 row(s) returned	0.000 sec / 0.000 sec
32	14:41:09	SELECT EMP_ID,FIRST_NAME, LAST_NAME,GENDER, DEPT FROM emp_record_table...	19 row(s) returned	0.016 sec / 0.000 sec
33	14:43:03	SELECT EMP_ID,FIRST_NAME, LAST_NAME,GENDER, DEPT,EMP_RATING FROM em...	3 row(s) returned	0.000 sec / 0.000 sec

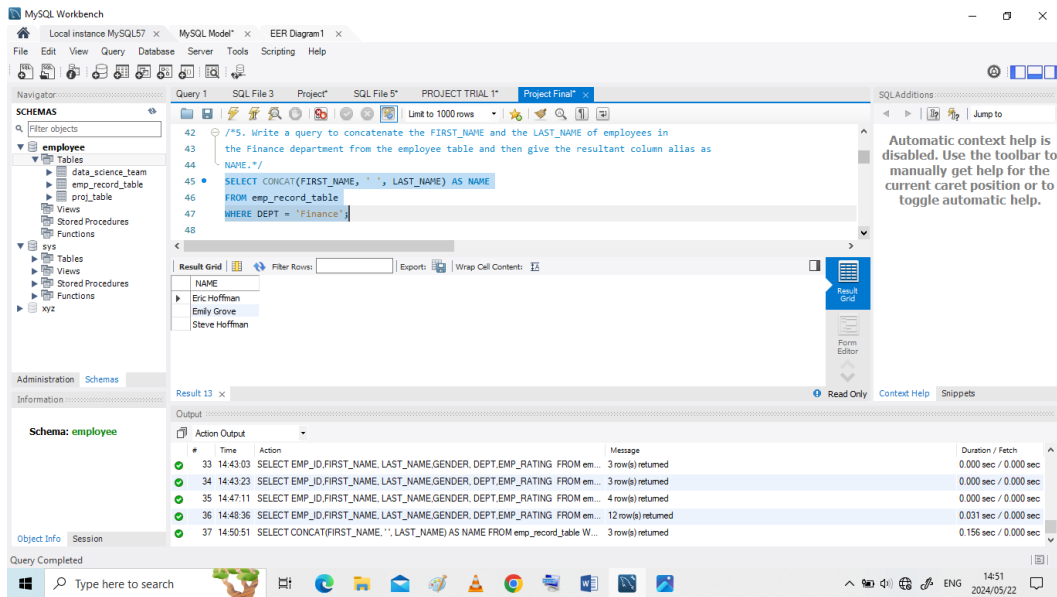


- Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS NAME
```

```
FROM emp_record_table
```

```
WHERE DEPT = 'Finance';
```



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

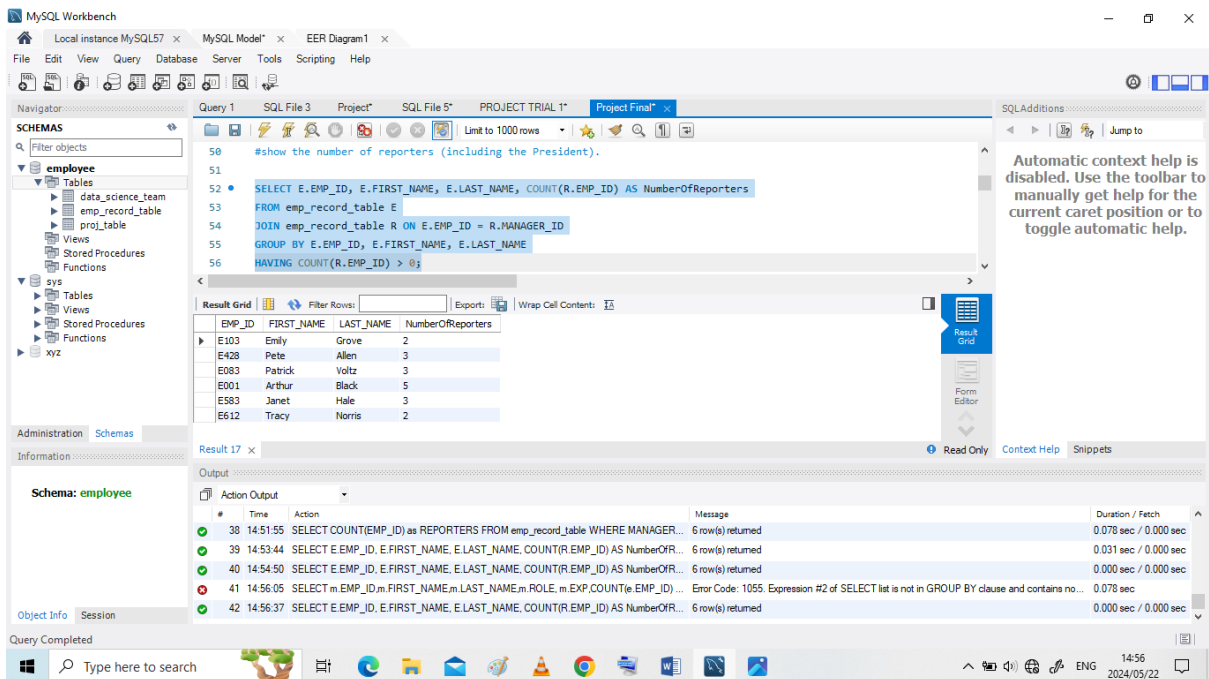
```
SELECT E.EMP_ID, E.FIRST_NAME, E.LAST_NAME, COUNT(R.EMP_ID) AS
NumberOfReporters
```

```
FROM emp_record_table E
```

```
JOIN emp_record_table R ON E.EMP_ID = R.MANAGER_ID
```

```
GROUP BY E.EMP_ID, E.FIRST_NAME, E.LAST_NAME
```

```
HAVING COUNT(R.EMP_ID) > 0;
```

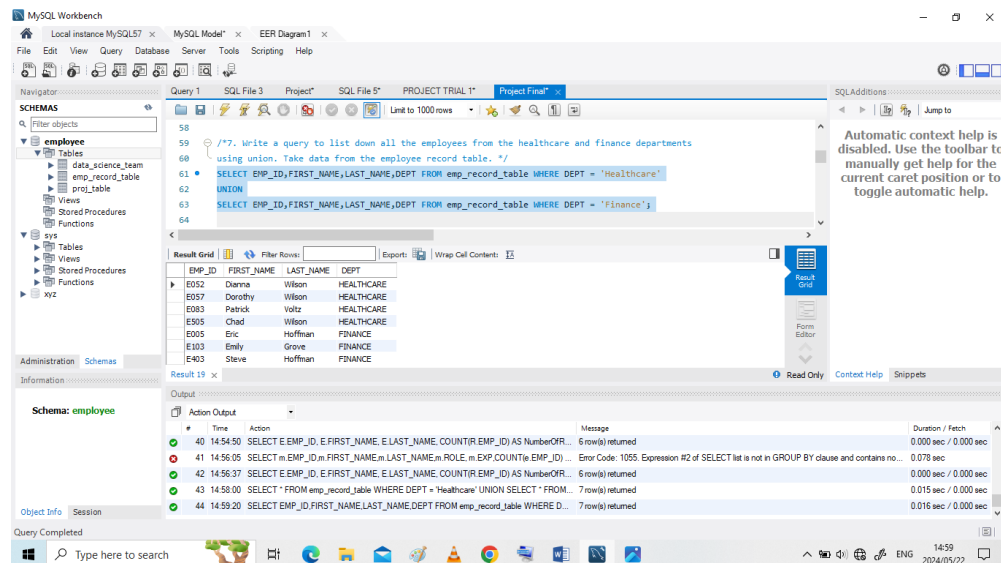


7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.

```
SELECT EMP_ID,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';
```



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 DEPT, EMP_ID, FIRST_NAME, LAST_NAME, ROLE, EMP_RATING,  
MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MaxEmpRating
```

```
FROM emp_record_table;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

64
65 /*8. Write a query to list down employee details such as EMP_ID, FIRST_NAME,
66 LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING grouped by dept. Also include the
67 respective employee rating along with the max emp rating for the department. */
68 SELECT DEPT, EMP_ID, FIRST_NAME, LAST_NAME, ROLE, EMP_RATING, MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MaxEmpRating
69 FROM emp_record_table;
70

```

DEPT	EMP_ID	FIRST_NAME	LAST_NAME	ROLE	EMP_RATING	MaxEmpRating
ALL	E001	Arthur	Black	PRESIDENT	5	5
AUTOMOTIVE	E010	William	Butler	LEAD DATA SCIENTIST	2	5
AUTOMOTIVE	E204	Karene	Nowak	SENIOR DATA SCIENTIST	5	5
AUTOMOTIVE	E428	Pete	Allen	MANAGER	4	5
AUTOMOTIVE	E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	1	5
FINANCE	E005	Eric	Hoffman	LEAD DATA SCIENTIST	3	4
FINANCE	E103	Emily	Grove	MANAGER	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.

```

SELECT ROLE, MIN(SALARY) AS MinSalary, MAX(SALARY) AS MaxSalary
FROM emp_record_table
GROUP BY ROLE

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

71 /*9. Write a query to calculate the minimum and the maximum salary of the employees in
72 each role. Take data from the employee record table. */
73 SELECT ROLE, MIN(SALARY) AS MinSalary, MAX(SALARY) AS MaxSalary
74 FROM emp_record_table
75 GROUP BY ROLE;
76

```

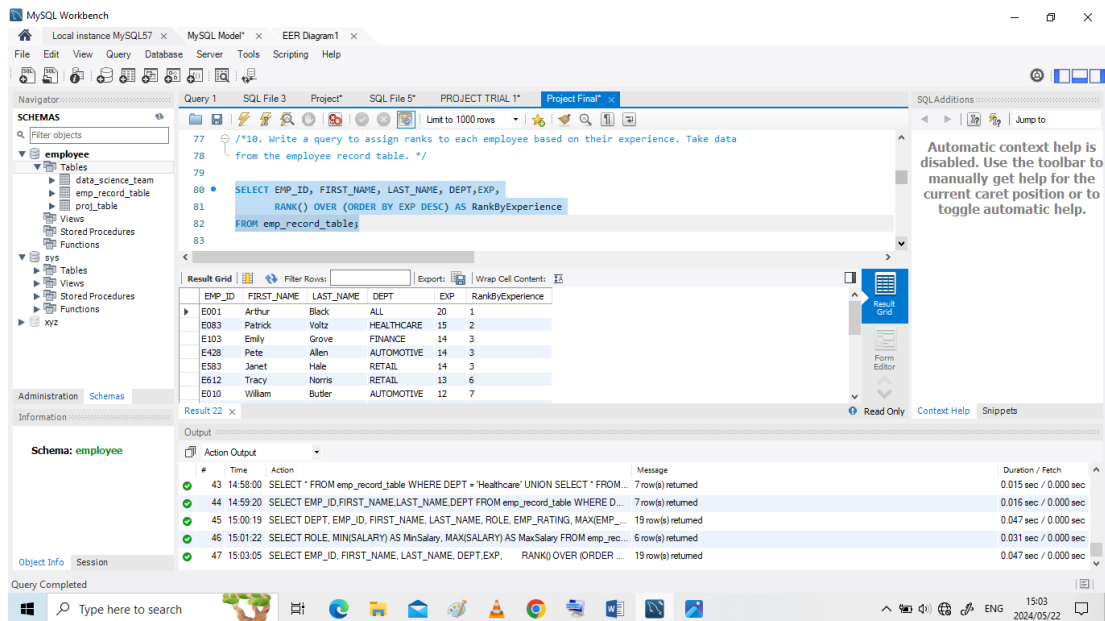
ROLE	MinSalary	MaxSalary
PRESIDENT	16500	16500
LEAD DATA SCIENTIST	8500	9000
SENIOR DATA SCIENTIST	9500	9700
MANAGER	8500	11000
ASSOCIATE DATA SCIENTIST	4000	5000
JUNIOR DATA SCIENTIST	2800	3000

10. 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, DEPT, EXP,
RANK() OVER (ORDER BY EXP DESC) AS RankByExperience
FROM 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.

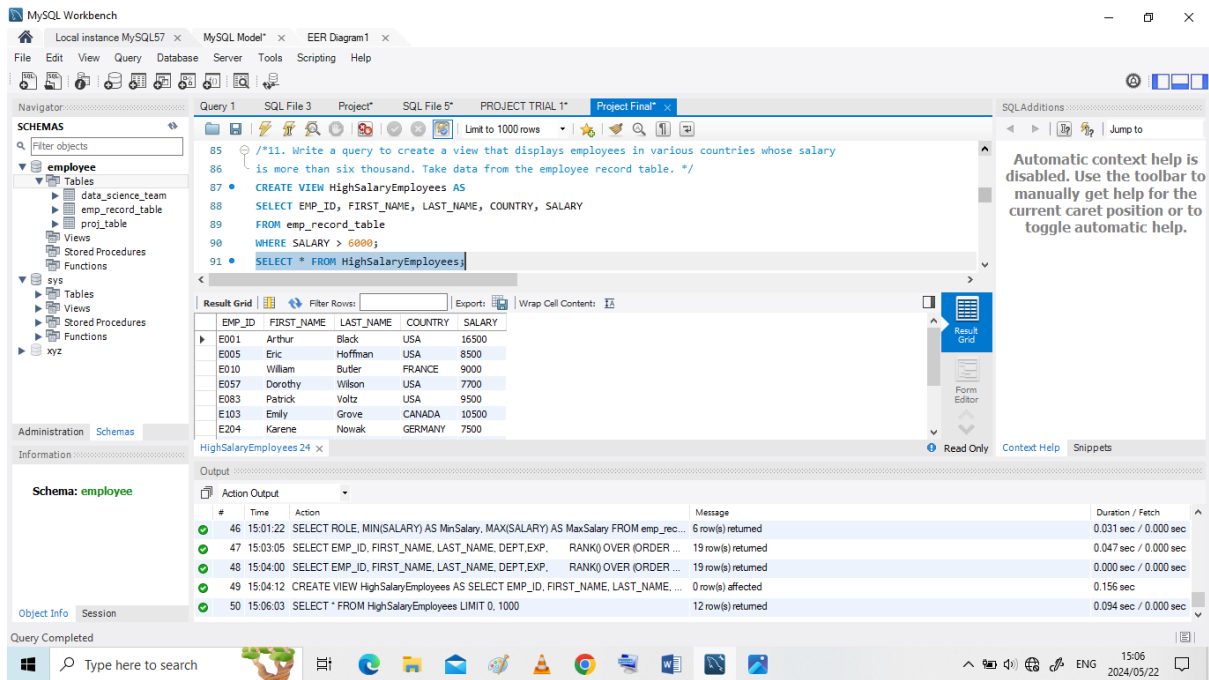
CREATE VIEW HighSalaryEmployees AS

SELECT EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY

FROM emp_record_table

WHERE SALARY > 6000;

SELECT * FROM HighSalaryEmployees;



12. 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

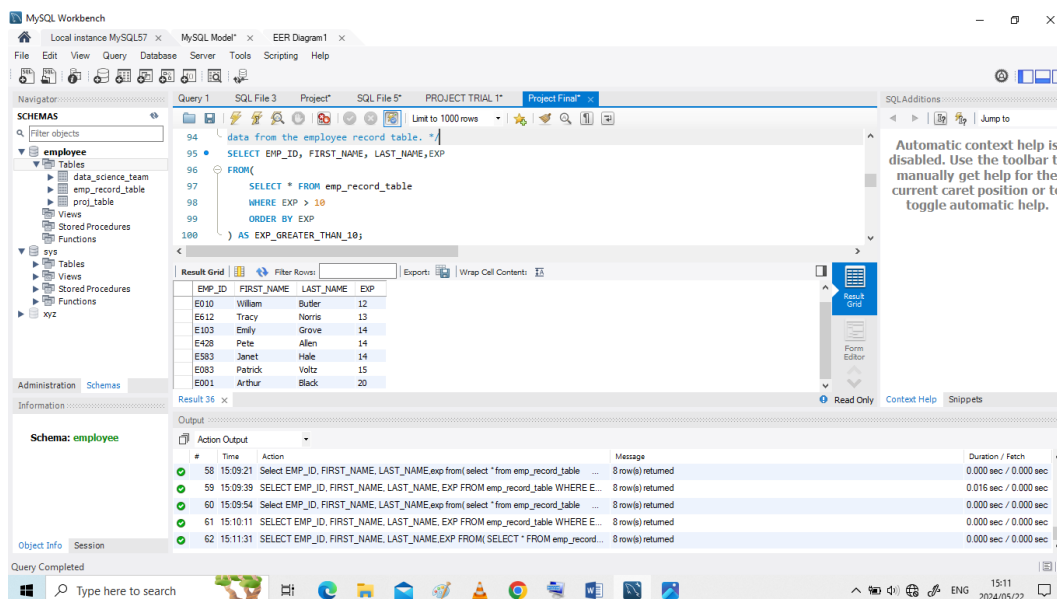
FROM(

SELECT * FROM emp_record_table

WHERE EXP > 10

ORDER BY EXP

) AS EXP_GREATER_THAN_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.

```
DELIMITER //
```

```
CREATE PROCEDURE GetExperiencedEmployees()
```

```
BEGIN
```

```
    SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP
```

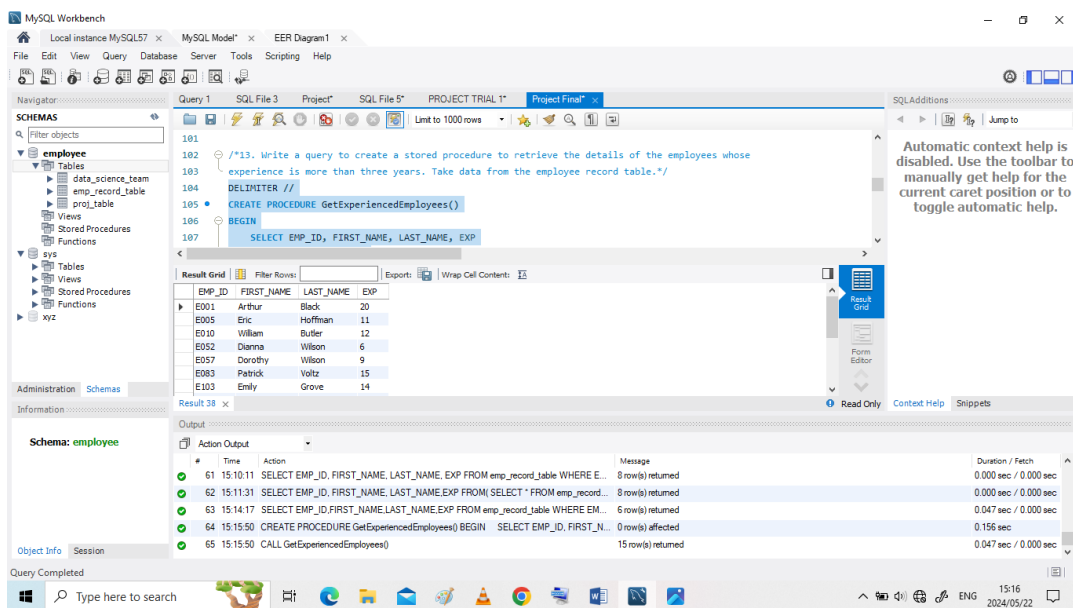
```
    FROM emp_record_table
```

```
    WHERE EXP > 3;
```

```
END //
```

```
DELIMITER ;
```

```
CALL GetExperiencedEmployees();
```



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.

```
DELIMITER &&
```

```
CREATE FUNCTION Employee_ROLE(
```

```
    EXP int
```

```
)
```

```
    RETURNS VARCHAR(40)
```

DETERMINISTIC

BEGIN

DECLARE Employee_ROLE VARCHAR(40);

IF EXP>12 AND 16 THEN

SET Employee_ROLE="MANAGER";

ELSEIF EXP>10 AND 12 THEN

SET Employee_ROLE ="LEAD DATA SCIENTIST";

ELSEIF EXP>5 AND 10 THEN

SET Employee_ROLE ="SENIOR DATA SCIENTIST";

ELSEIF EXP>2 AND 5 THEN

SET Employee_ROLE ="ASSOCIATE DATA SCIENTIST";

ELSEIF EXP<=2 THEN

SET Employee_ROLE ="JUNIOR DATA SCIENTIST";

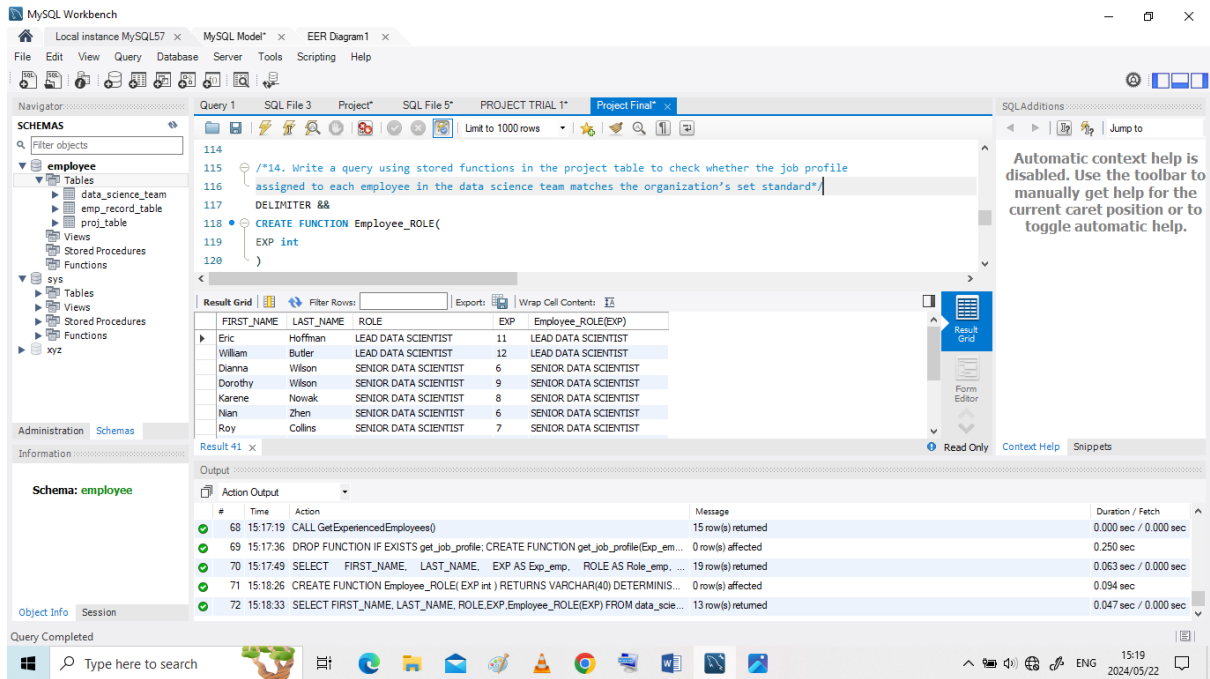
END IF;

RETURN (Employee_ROLE);

END &&

SELECT FIRST_NAME, LAST_NAME, ROLE,EXP,Employee_ROLE(EXP)

FROM data_science_team;

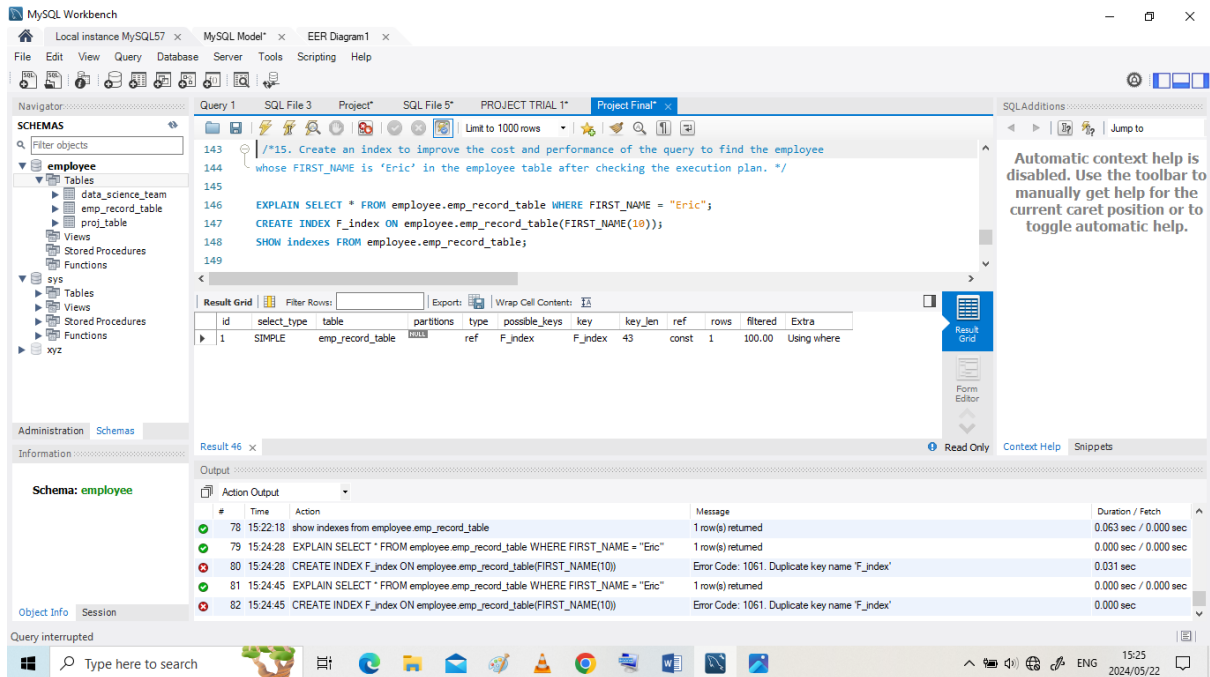


15. Create an index to improve the cost and performance of the query to find the employee whose FIRST_NAME is 'Eric' in the employee table after checking the execution plan.

```
EXPLAIN SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = "Eric";
```

```
CREATE INDEX F_index ON employee.emp_record_table(FIRST_NAME(10));
```

```
SHOW indexes FROM employee.emp_record_table;
```

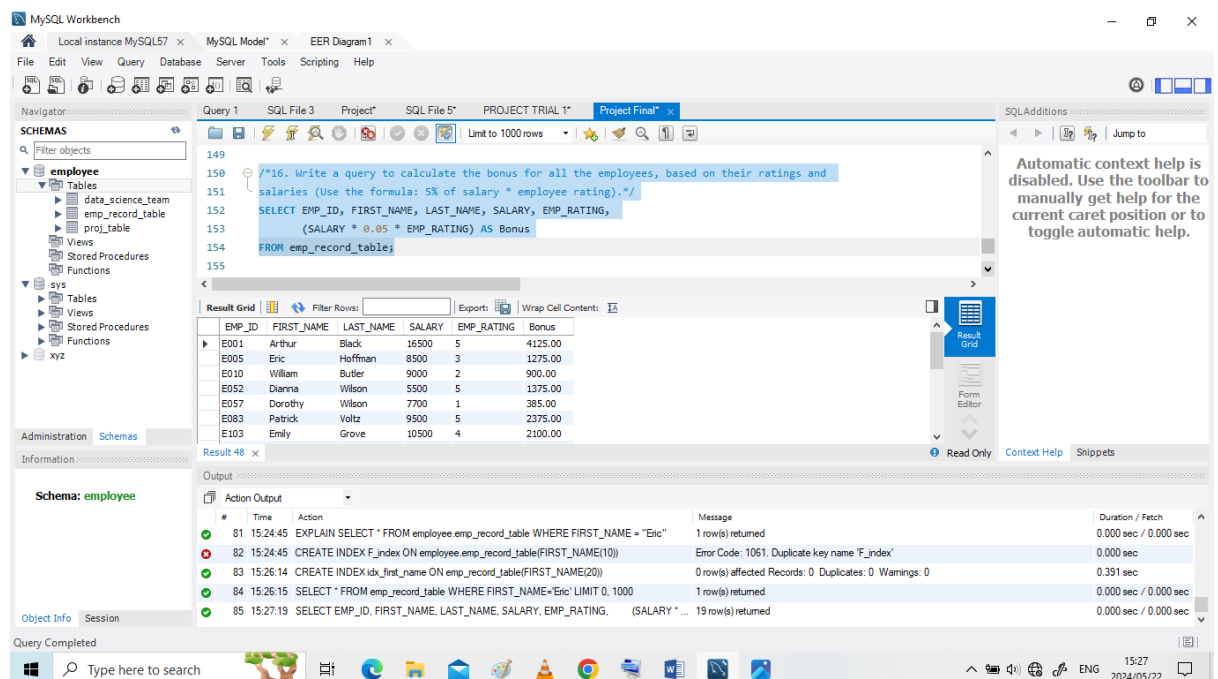


16. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary * employee rating).

```

SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING,
       (SALARY * 0.05 * EMP_RATING) AS Bonus
FROM 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.

```
SELECT CONTINENT, COUNTRY, AVG(SALARY) AS AverageSalary
```

```
FROM emp_record_table
```

```
GROUP BY CONTINENT, COUNTRY;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
155
156
157
158 SELECT CONTINENT, COUNTRY, AVG(SALARY) AS AverageSalary
159 FROM emp_record_table
160 GROUP BY CONTINENT, COUNTRY;
161
```

The Results Grid displays the following data:

CONTINENT	COUNTRY	AverageSalary
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

The Output pane shows the following messages:

```
89 15:31:04 SELECT CONTINENT, COUNTRY, AVG(SALARY) AS AverageSalary FROM emp_record_t... 7 row(s) returned 0.000 sec / 0.000 sec
90 15:31:38 SELECT CONTINENT, COUNTRY, AVG(SALARY) AS AverageSalary FROM emp_record_t... Error Code: 1055. Expression #2 of SELECT list is not in GROUP BY clause and contains no... 0.000 sec
91 15:31:48 SELECT CONTINENT, COUNTRY, AVG(SALARY) AS AverageSalary FROM emp_record_t... 7 row(s) returned 0.000 sec / 0.000 sec
92 15:32:19 select CONTINENT, avg(SALARY) from emp_record_table group by CONTINENT order by ... 4 row(s) returned 0.000 sec / 0.000 sec
93 15:33:37 SELECT CONTINENT, COUNTRY, AVG(SALARY) AS AverageSalary FROM emp_record_t... 7 row(s) returned 0.000 sec / 0.000 sec
```