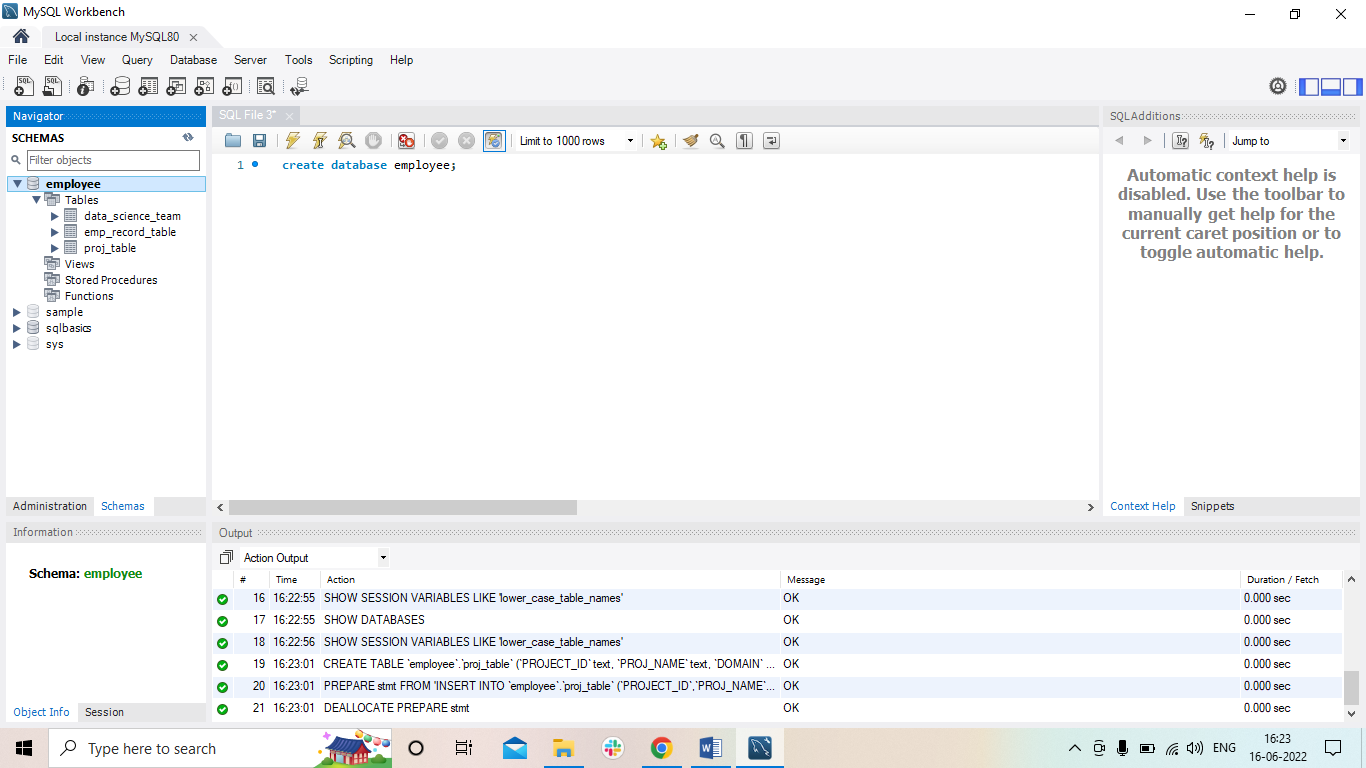
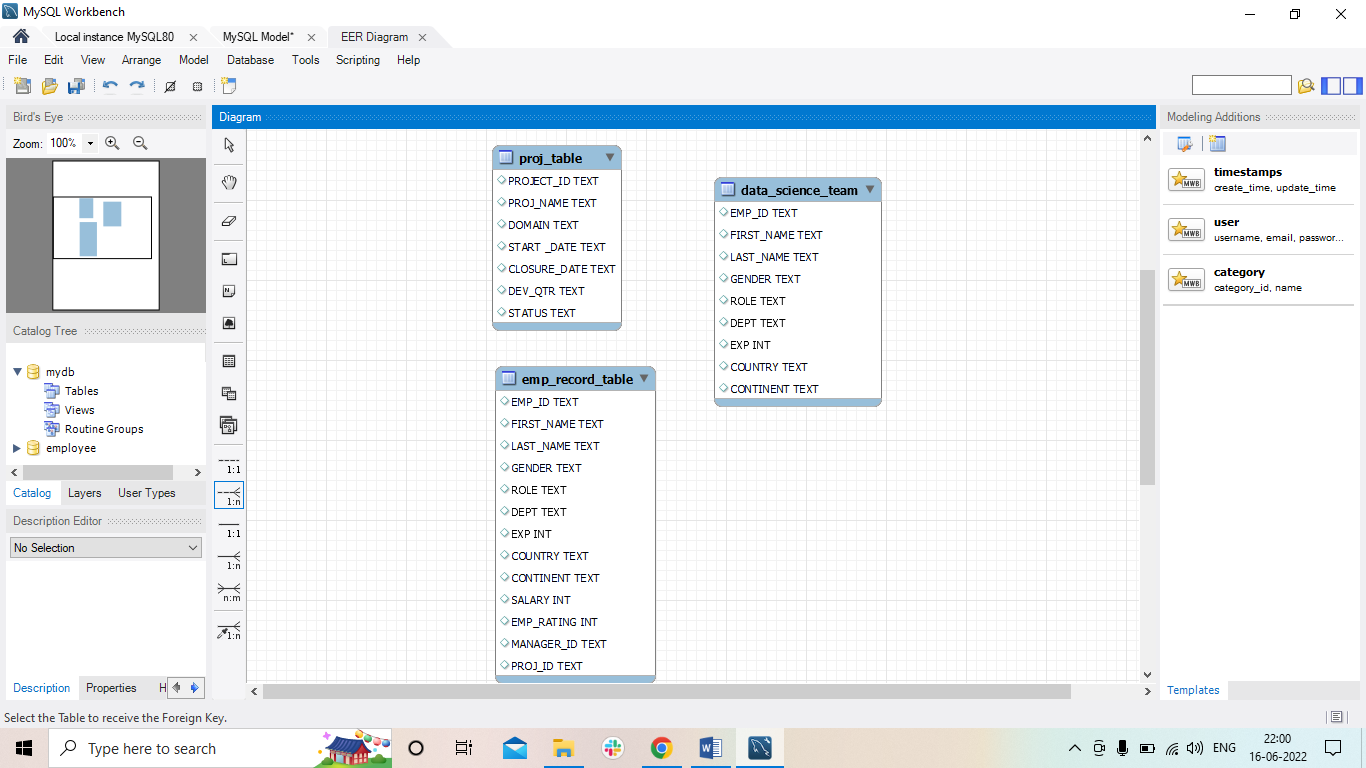
**ScienceQtech Employee Performance Mapping.**

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.



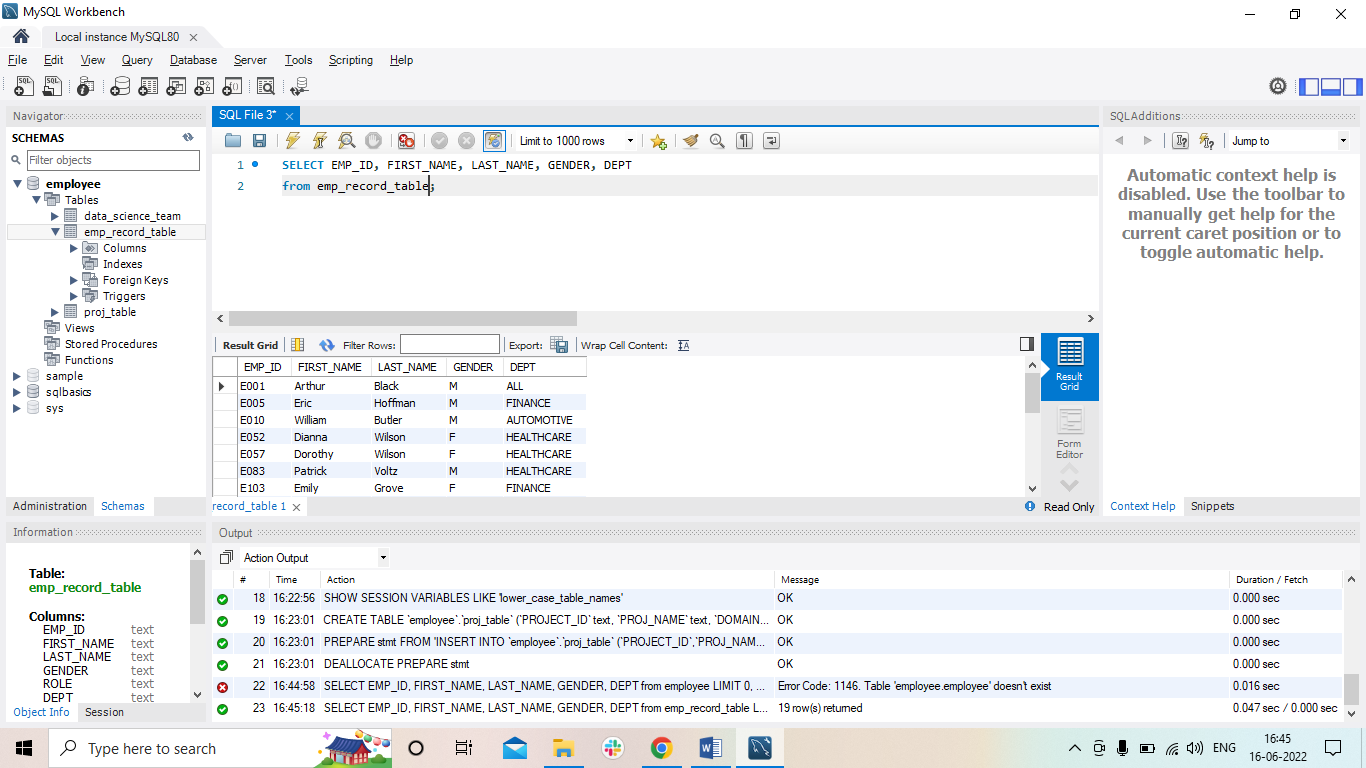
1. Create an ER diagram for the given **employee**database.



1. 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;



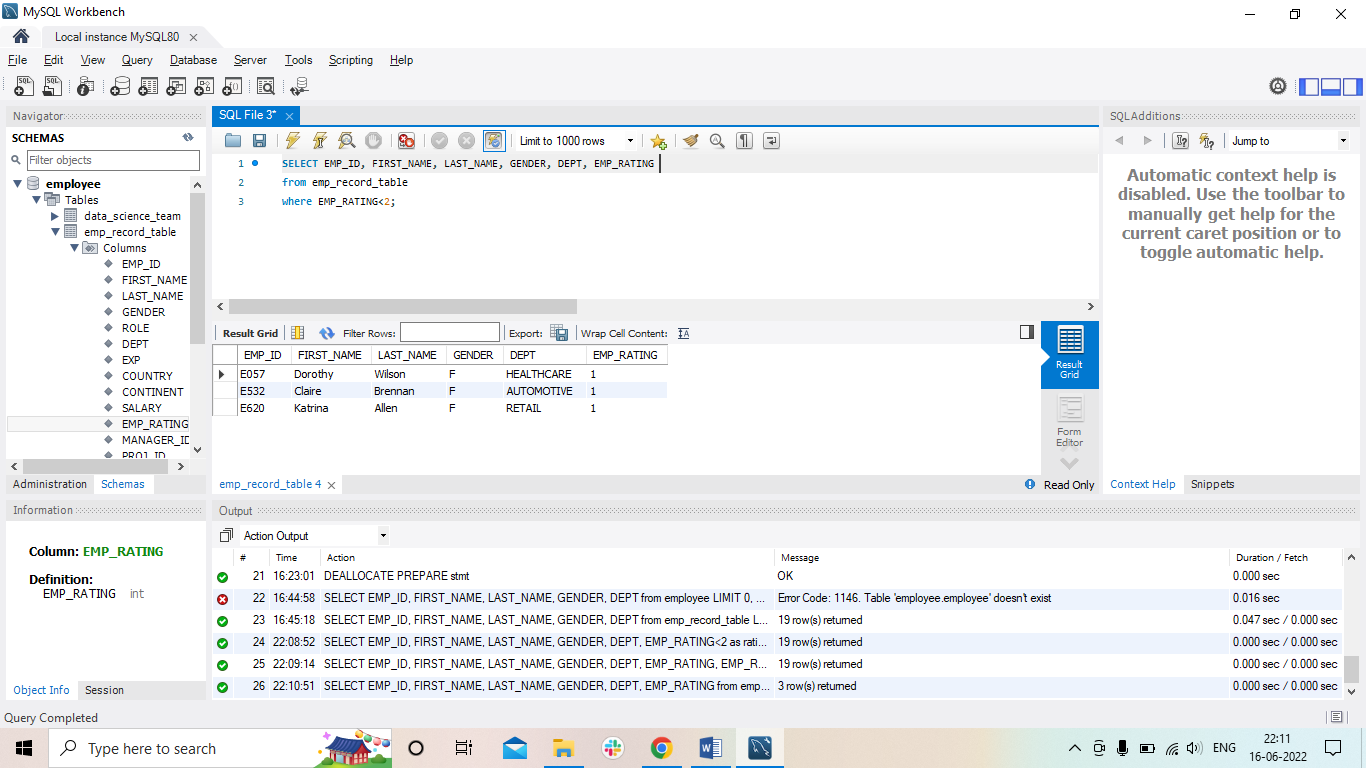
1. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:

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

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

from emp\_record\_table

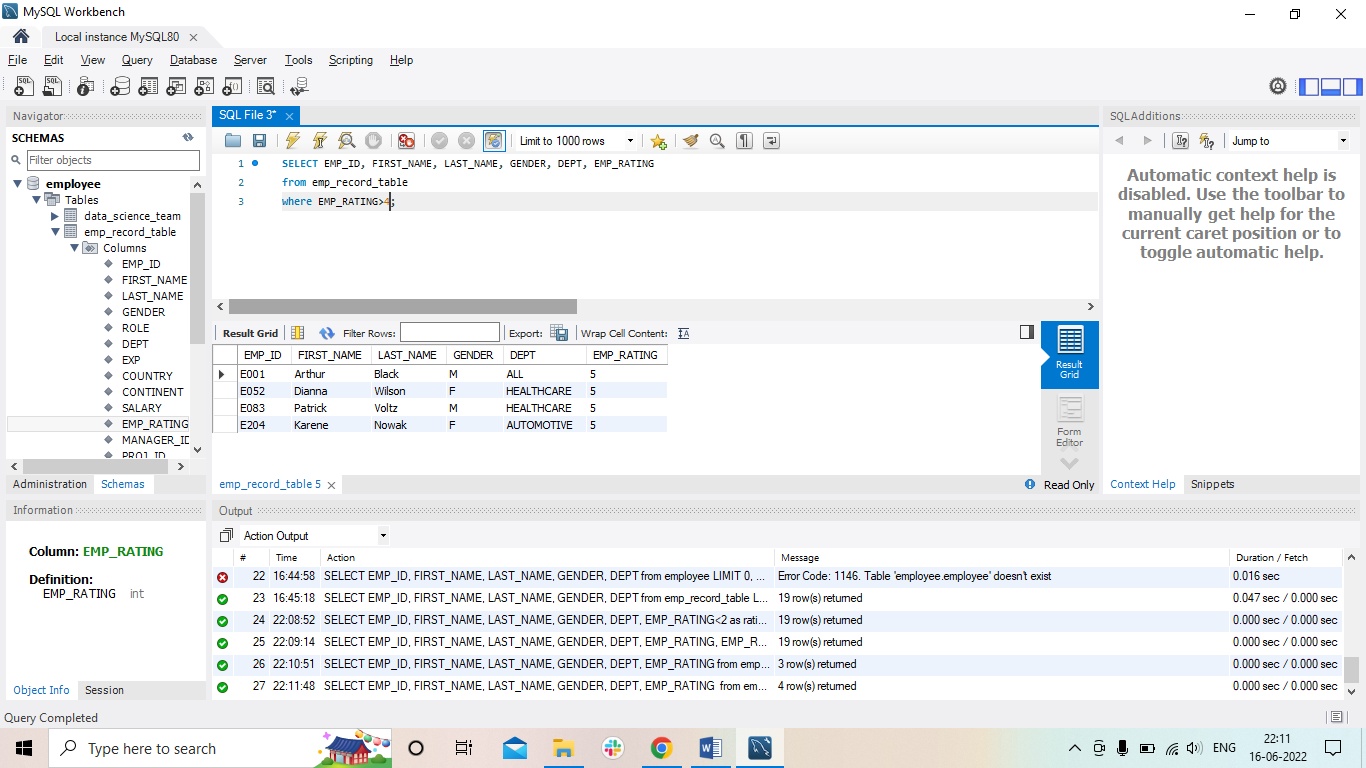
where EMP\_RATING<2;



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

from emp\_record\_table

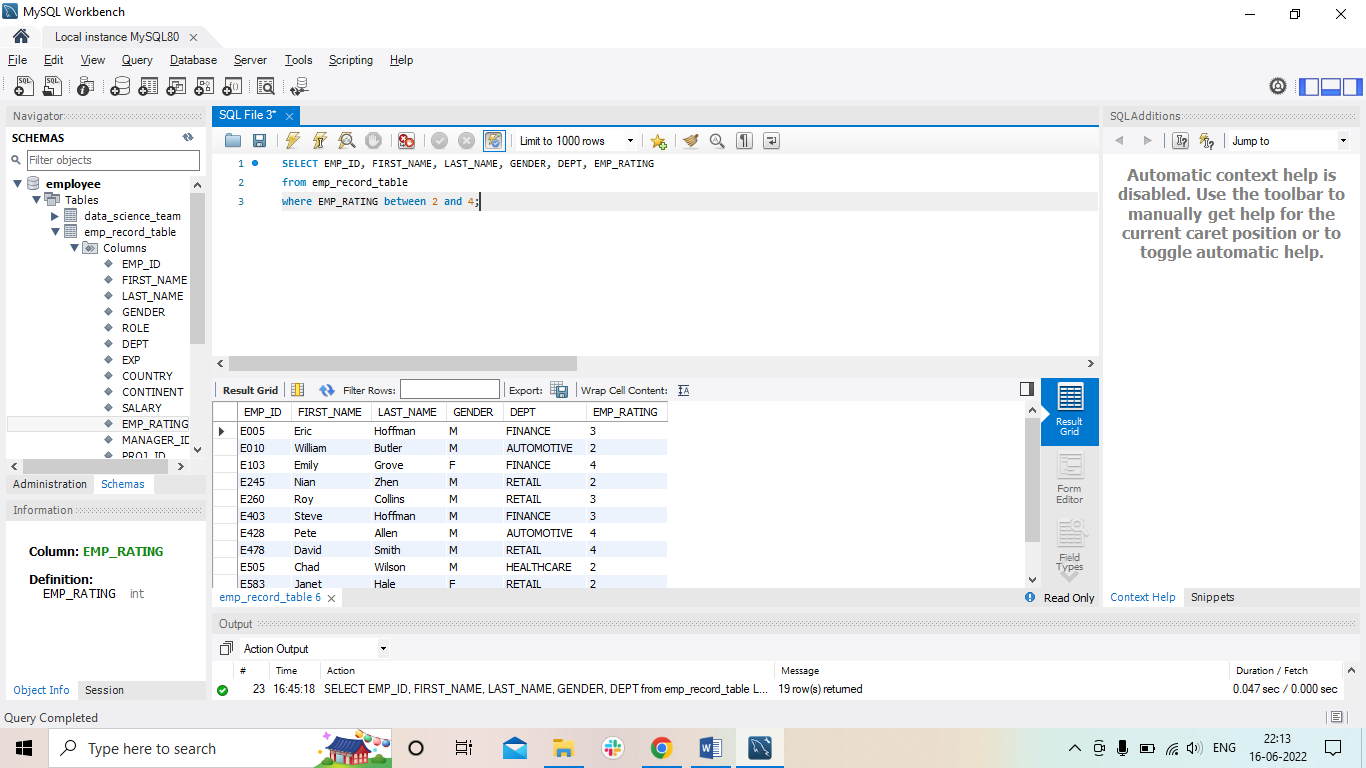
where EMP\_RATING>4;



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

from emp\_record\_table

where EMP\_RATING between 2 and 4;

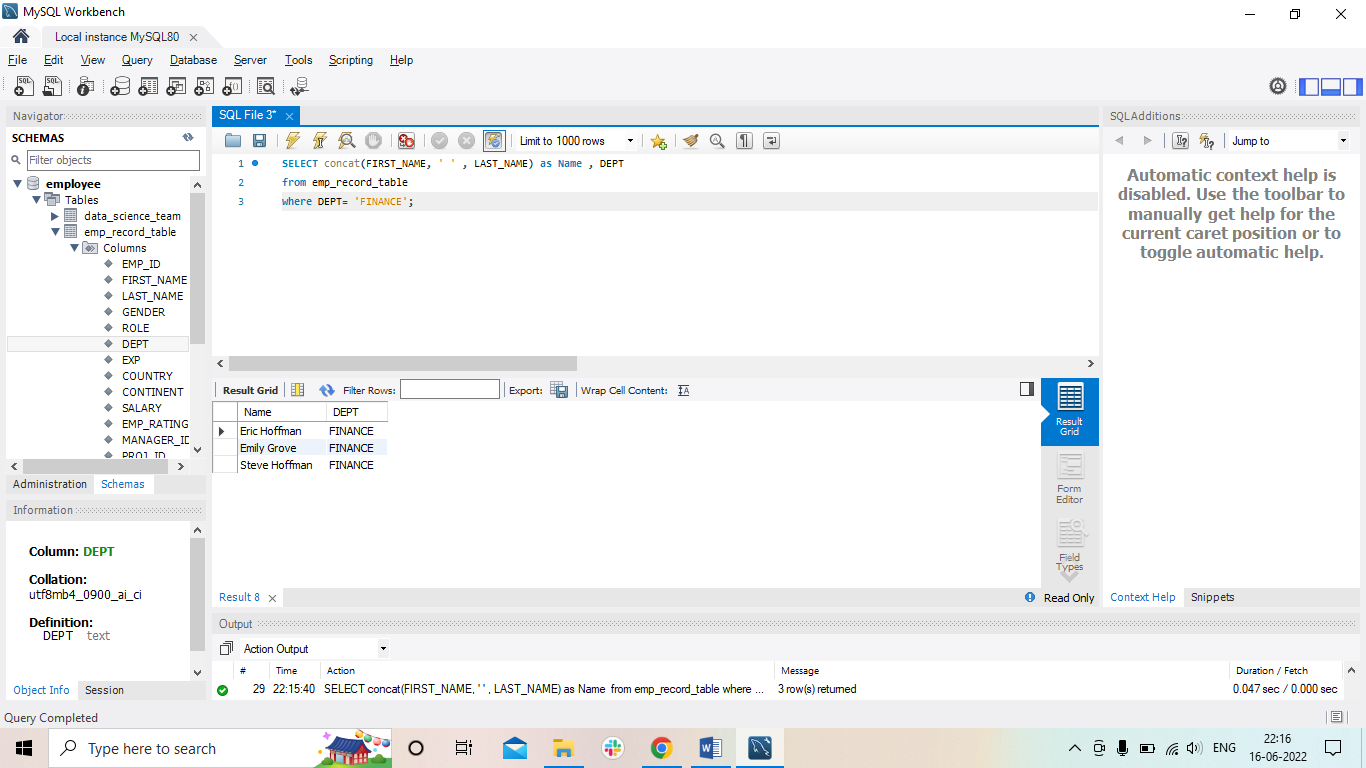


1. Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

SELECT concat(FIRST\_NAME, ' ' , LAST\_NAME) as Name , DEPT

from emp\_record\_table

where DEPT= 'FINANCE';

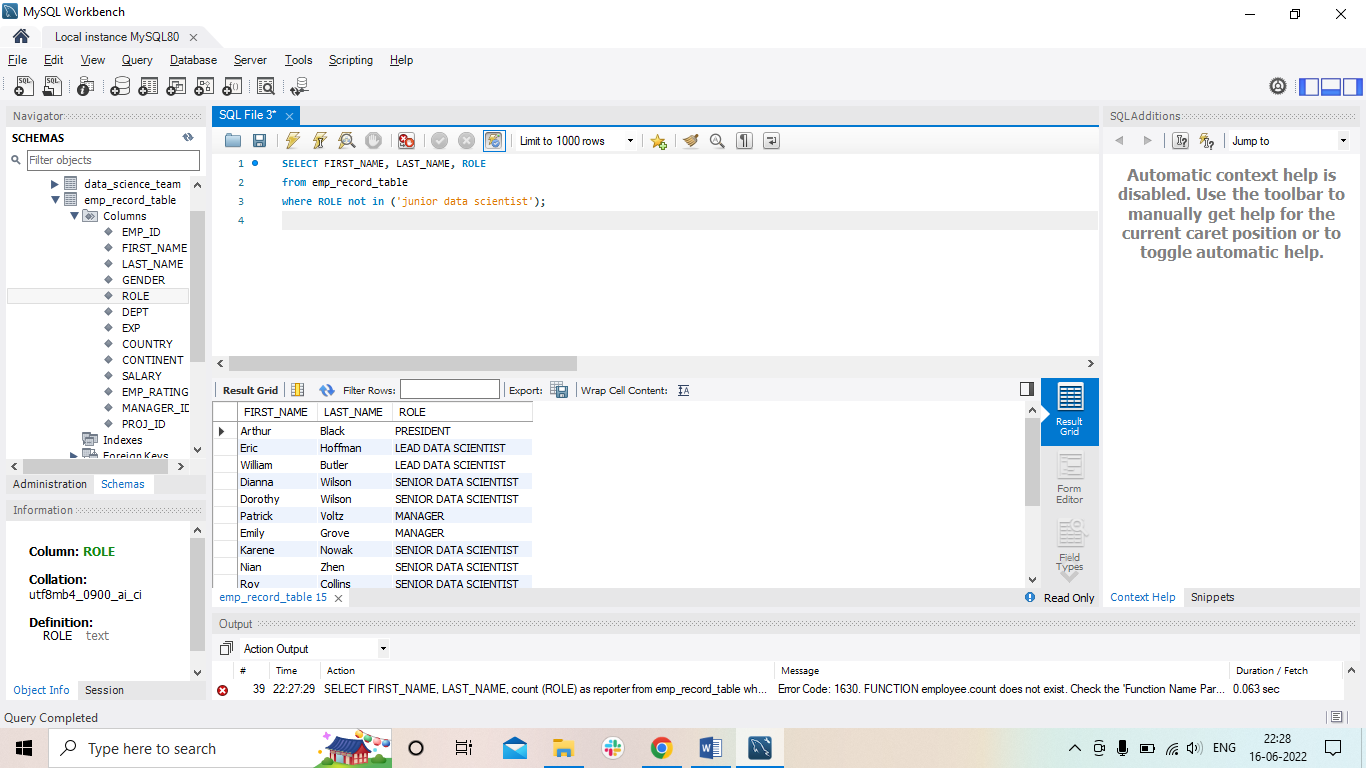


1. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).

SELECT FIRST\_NAME, LAST\_NAME, ROLE

from emp\_record\_table

where ROLE not in ('junior data scientist');



1. 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 \*

from emp\_record\_table

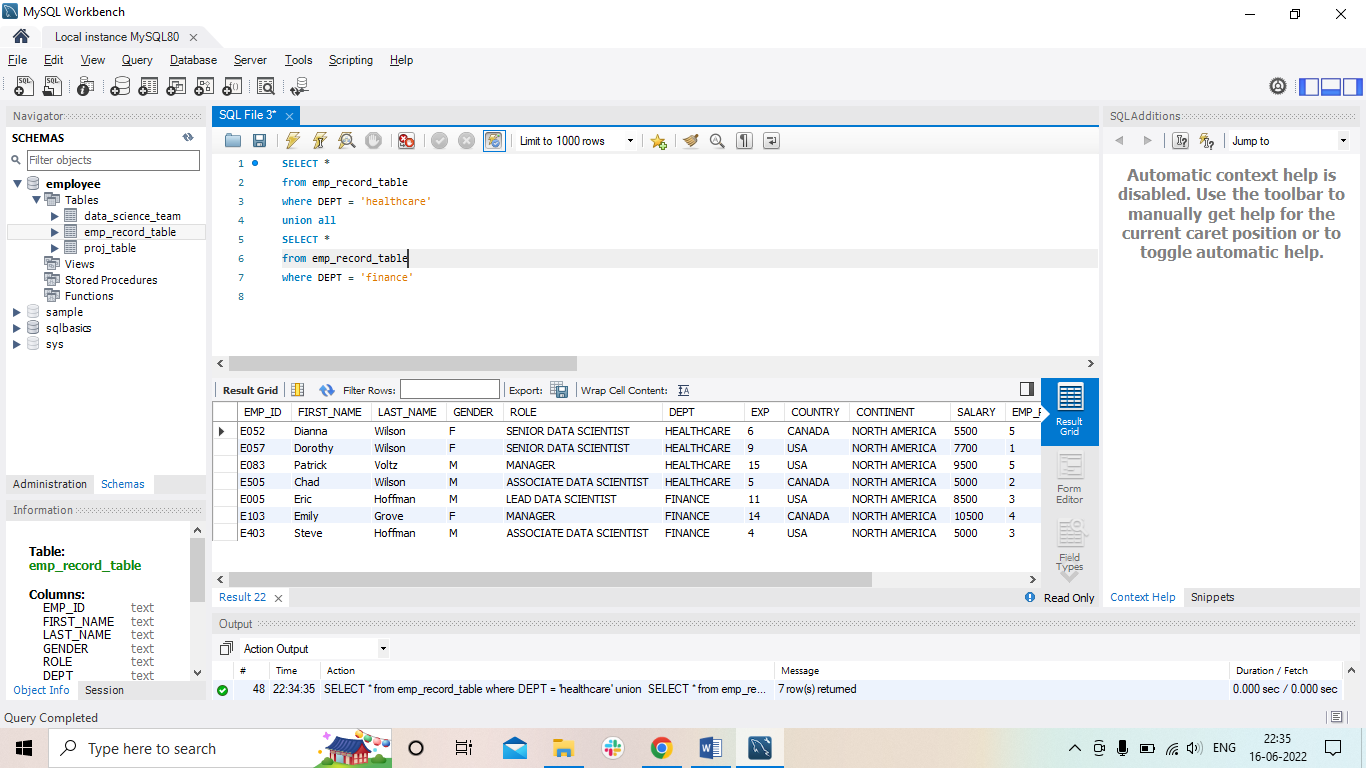
where DEPT = 'healthcare'

union all

SELECT \*

from emp\_record\_table

where DEPT = 'finance'



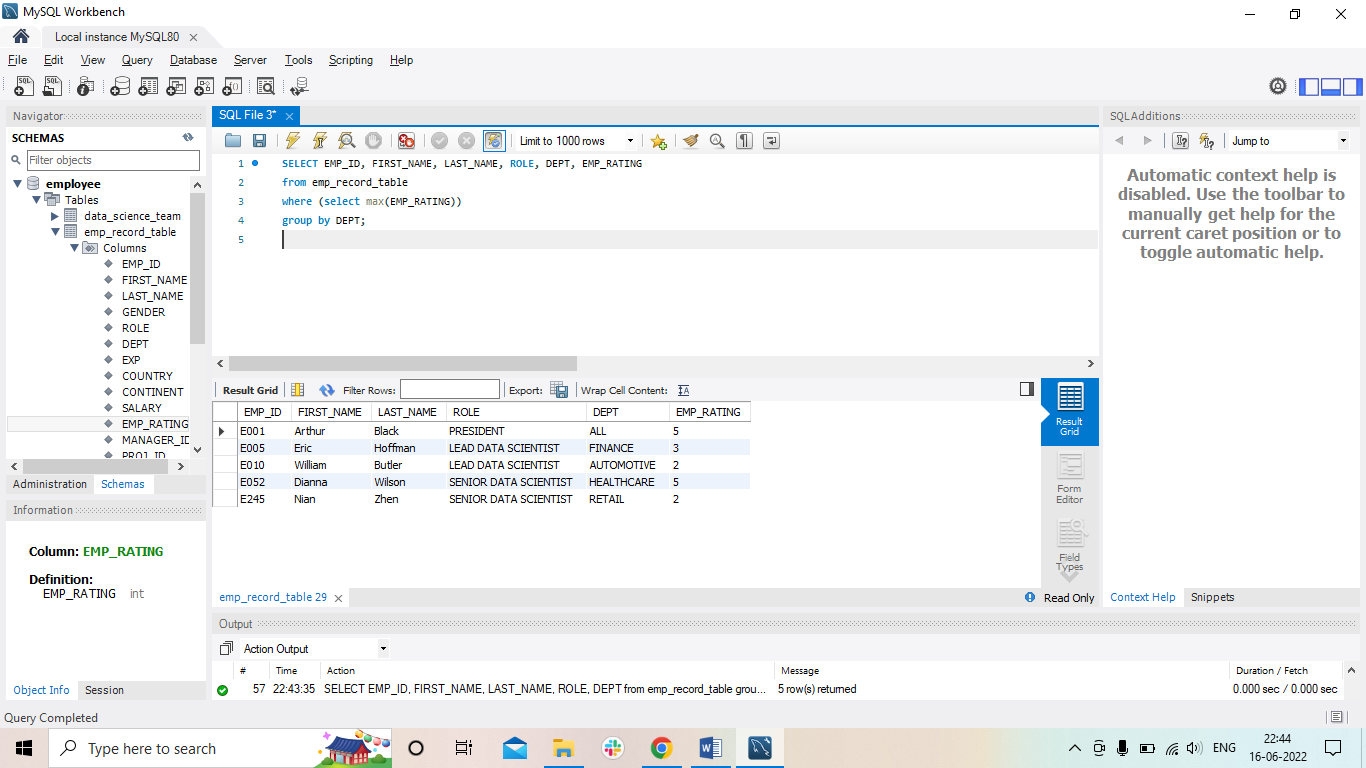
1. Write a query to list down employee details such as EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPARTMENT, and EMP\_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department.

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

from emp\_record\_table

where (select max(EMP\_RATING))

group by DEPT;

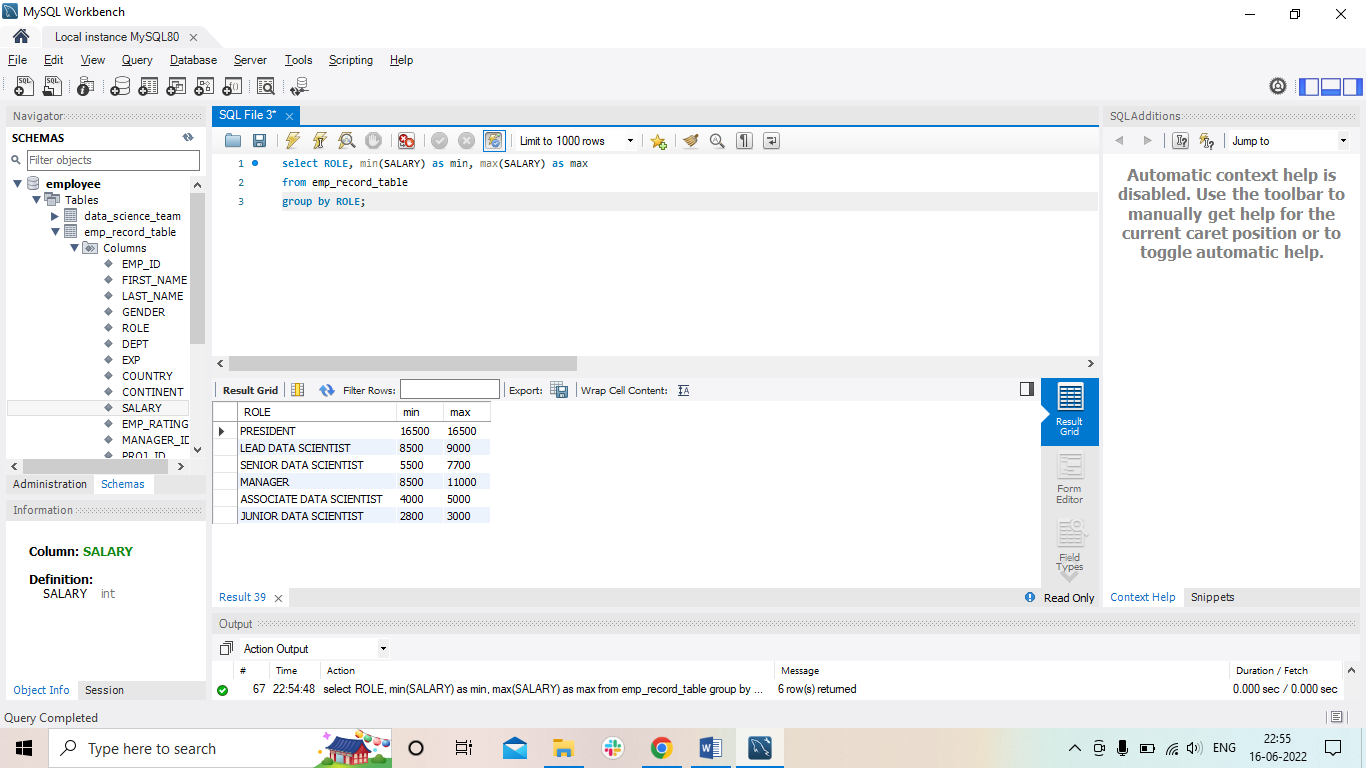


1. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

select ROLE, min(SALARY) as min, max(SALARY) as max

from emp\_record\_table

group by ROLE;

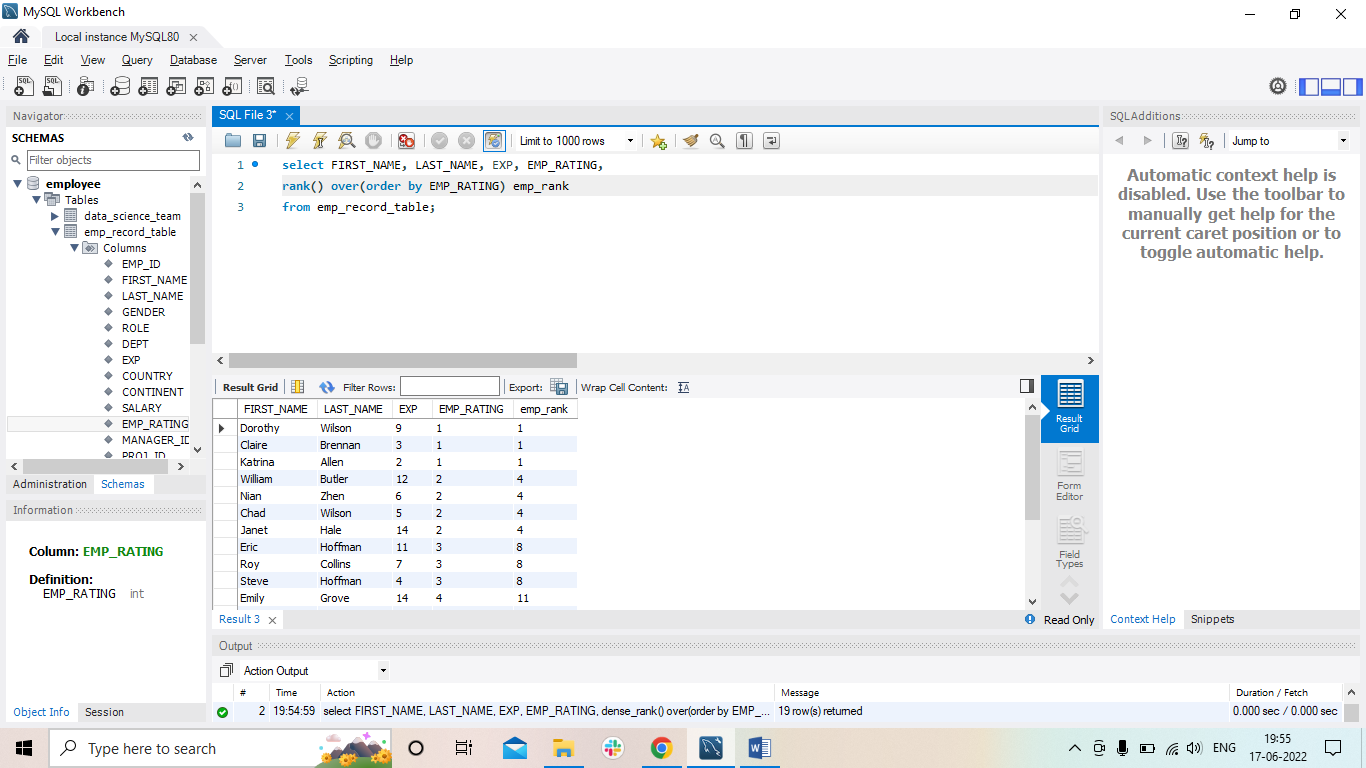


1. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

select FIRST\_NAME, LAST\_NAME, EXP, EMP\_RATING,

rank() over(order by EMP\_RATING) emp\_rank

from emp\_record\_table;



1. 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 eploye\_salary as

SELECT EMP\_ID, FIRST\_NAME, COUNTRY, SALARY

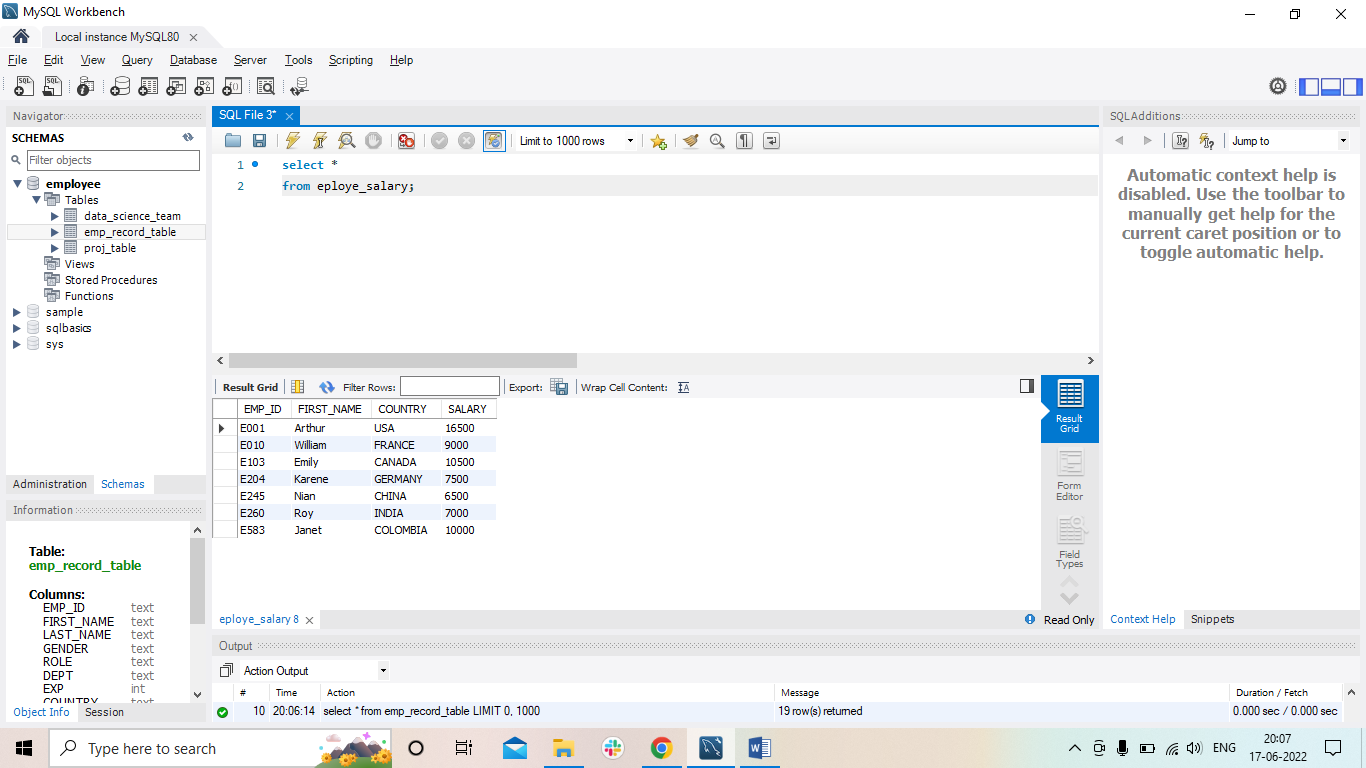
from emp\_record\_table

where SALARY > 6000

group by COUNTRY;

select \*

from eploye\_salary;

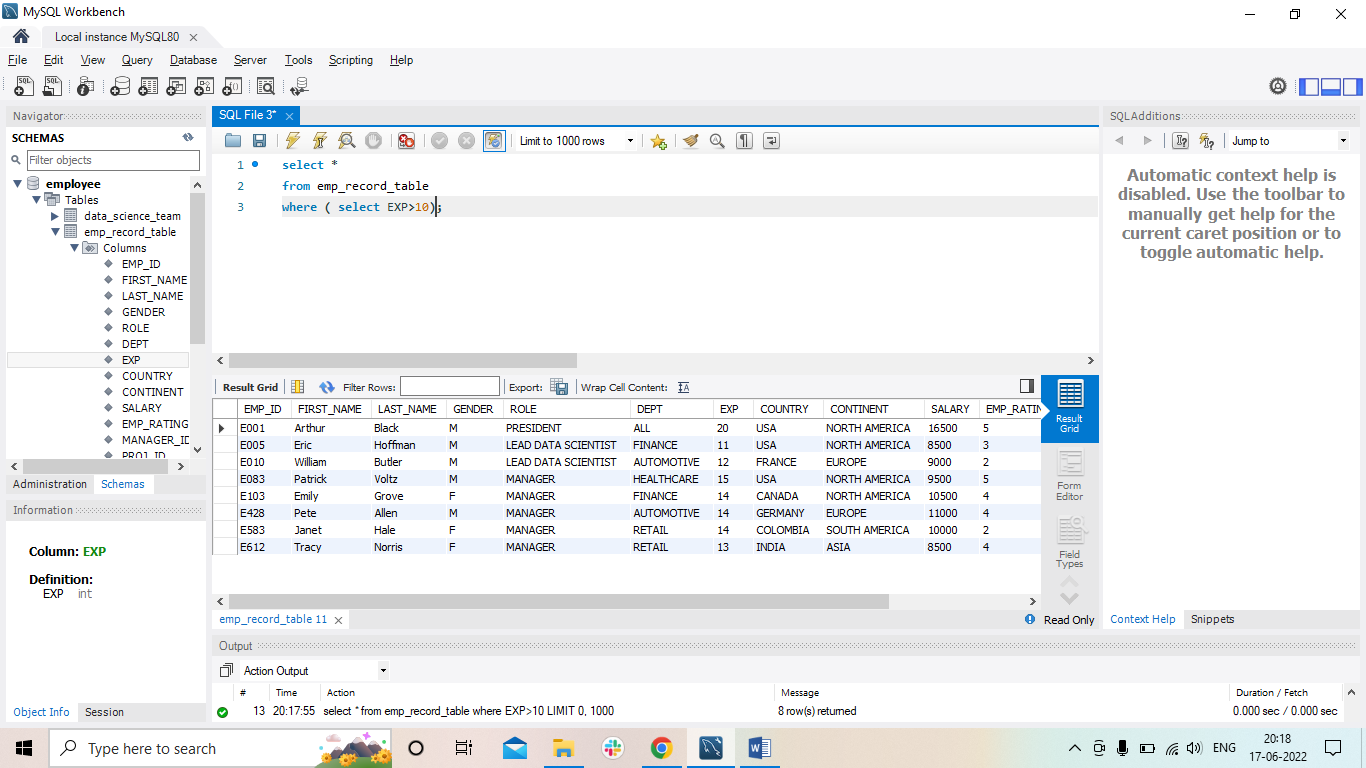


1. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

select \*

from emp\_record\_table

where ( select EXP>10);



1. 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 experience\_3more ()

Begin

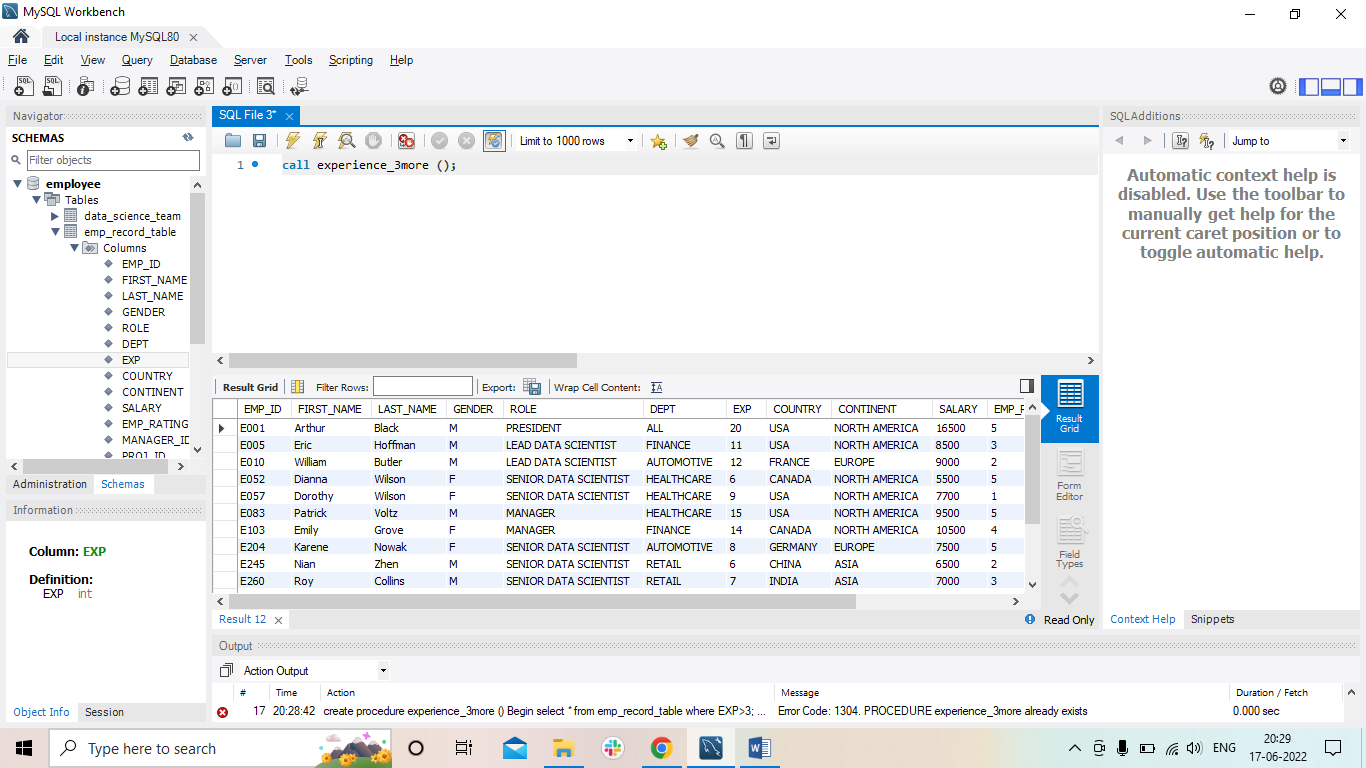
select \*

from emp\_record\_table

where EXP>3;

end &&

call experience\_3more ();



1. 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'.

drop procedure if exists roll\_experience;

delimiter &&

create procedure roll\_experience ( in expri int, out ROLEnew varchar(50))

Begin

select EXP, ROLE, DEPT

from data\_science\_team

where EXP=expri;

if EXP<=2 then

set ROLEnew = "'JUNIOR DATA SCIENTIST";

elseif EXP in (2 and 5) then

set ROLEnew = "ASSOCIATE DATA SCIENTIST";

elseif EXP in (5 and 10) then

set ROLEnew = "SENIOR DATA SCIENTIST";

elseif EXP in (10 and 12) then

set ROLEnew = "LEAD DATA SCIENTIST";

elseif EXP in (12 and 16) then

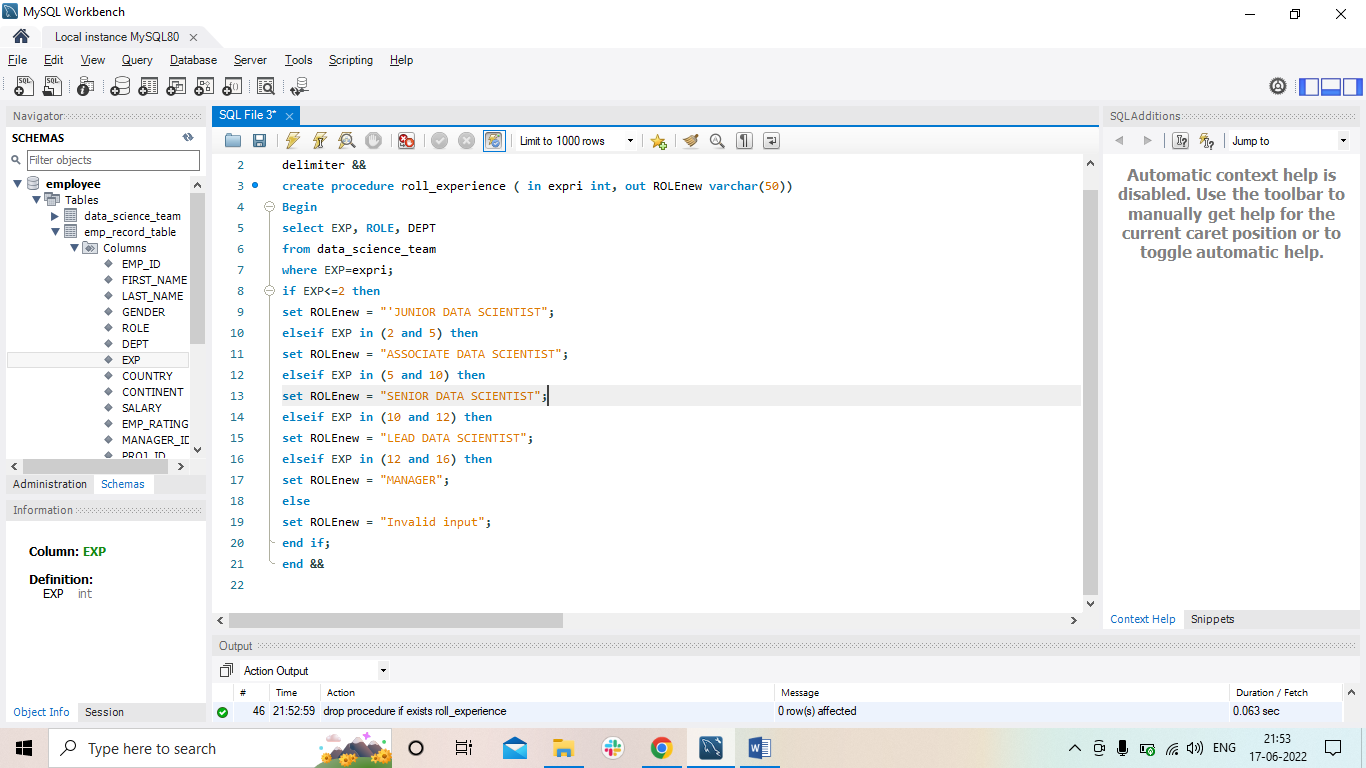
set ROLEnew = "MANAGER";

else

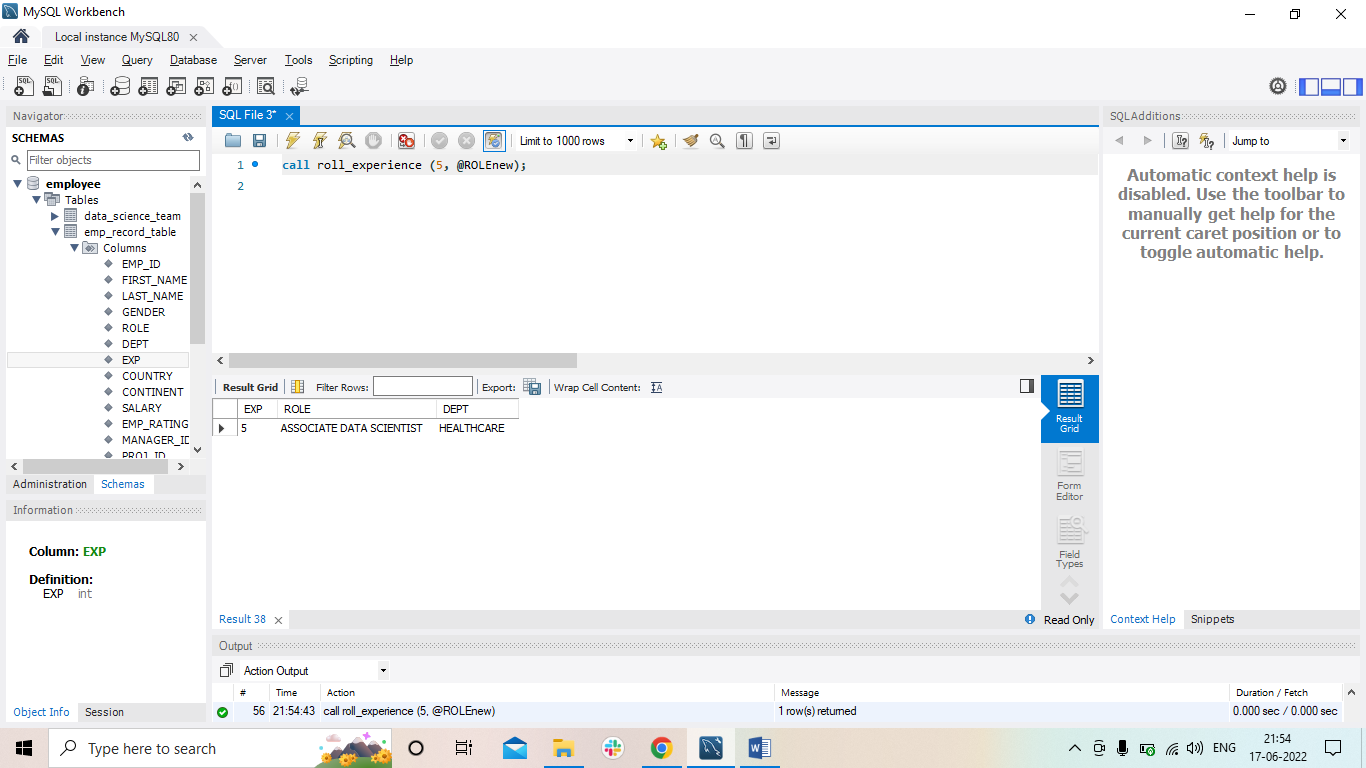
set ROLEnew = "Invalid input";

end if;

end &&



call roll\_experience (5, @ROLEnew);

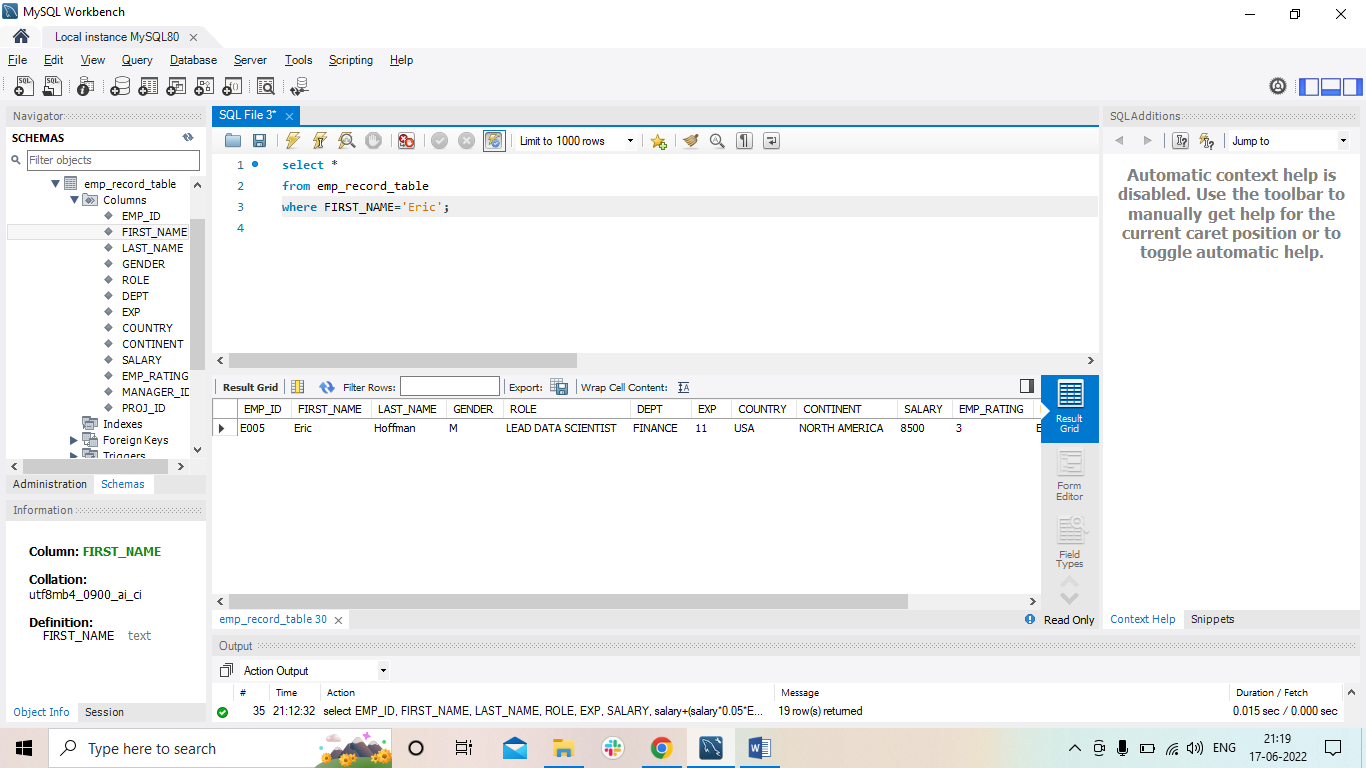


1. Create an index to improve the cost and performance of the query to find the employee whose FIRST\_NAME is ‘Eric’ in the employee table after checking the execution plan.

select \*

from emp\_record\_table

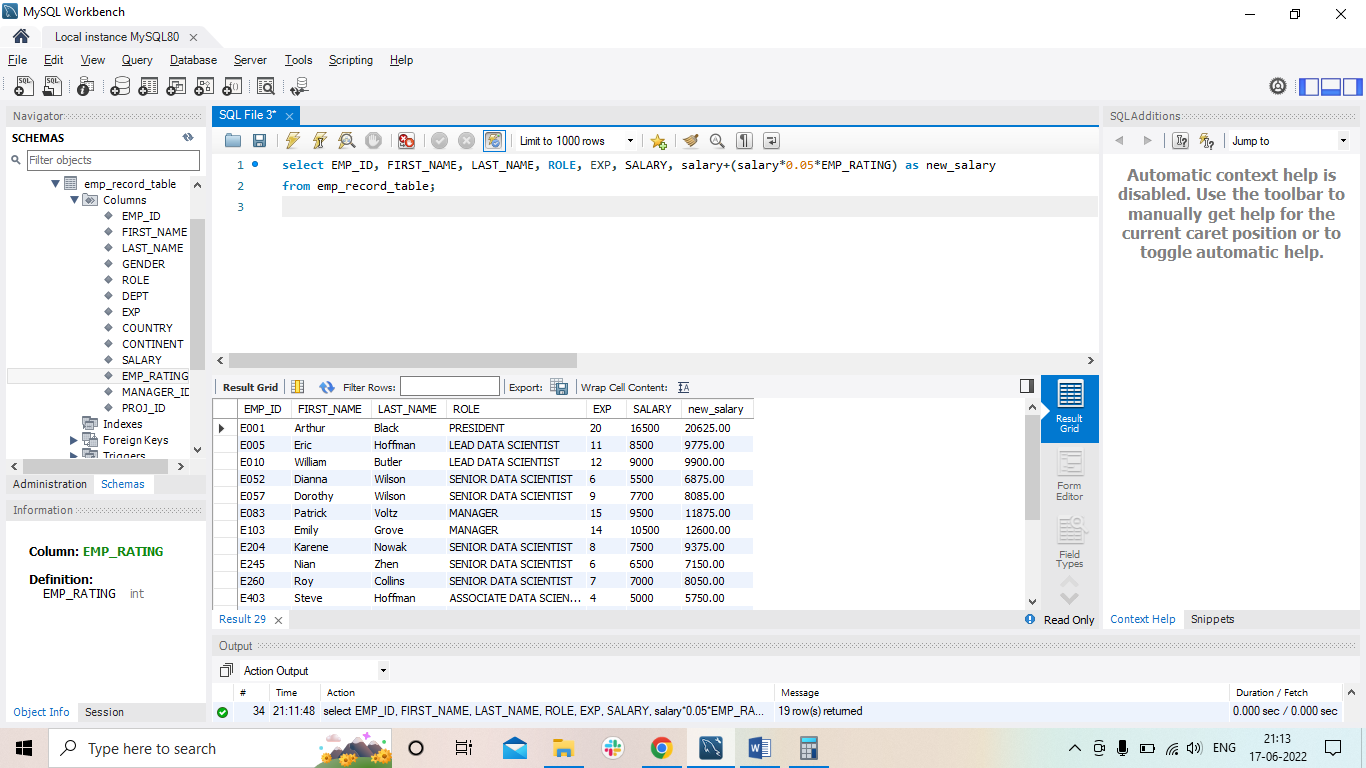
where FIRST\_NAME='Eric';



1. 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, ROLE, EXP, SALARY, salary+(salary\*0.05\*EMP\_RATING) as new\_salary

from emp\_record\_table;



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

select EMP\_ID, FIRST\_NAME, LAST\_NAME, COUNTRY, CONTINENT, ROLE, EXP, SALARY, avg(SALARY) as avg\_salary

from emp\_record\_table

group by CONTINENT

union

select EMP\_ID, FIRST\_NAME, LAST\_NAME, COUNTRY, CONTINENT, ROLE, EXP, SALARY, avg(SALARY) as avg\_salary

from emp\_record\_table

group by COUNTRY;

