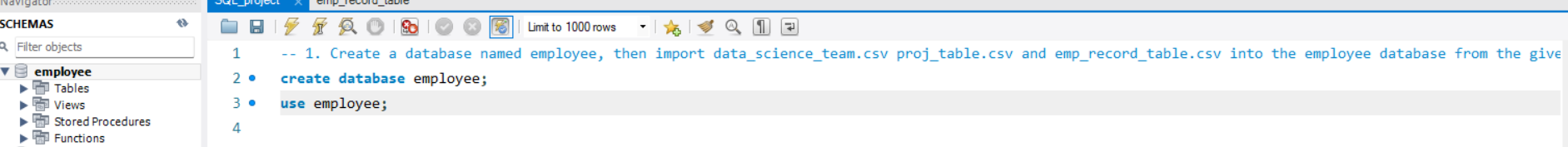
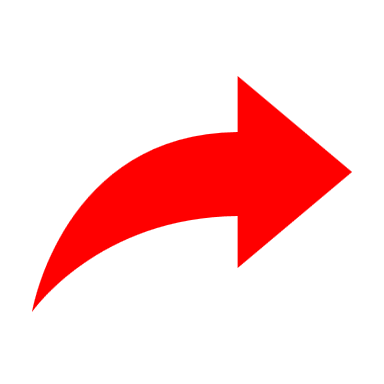
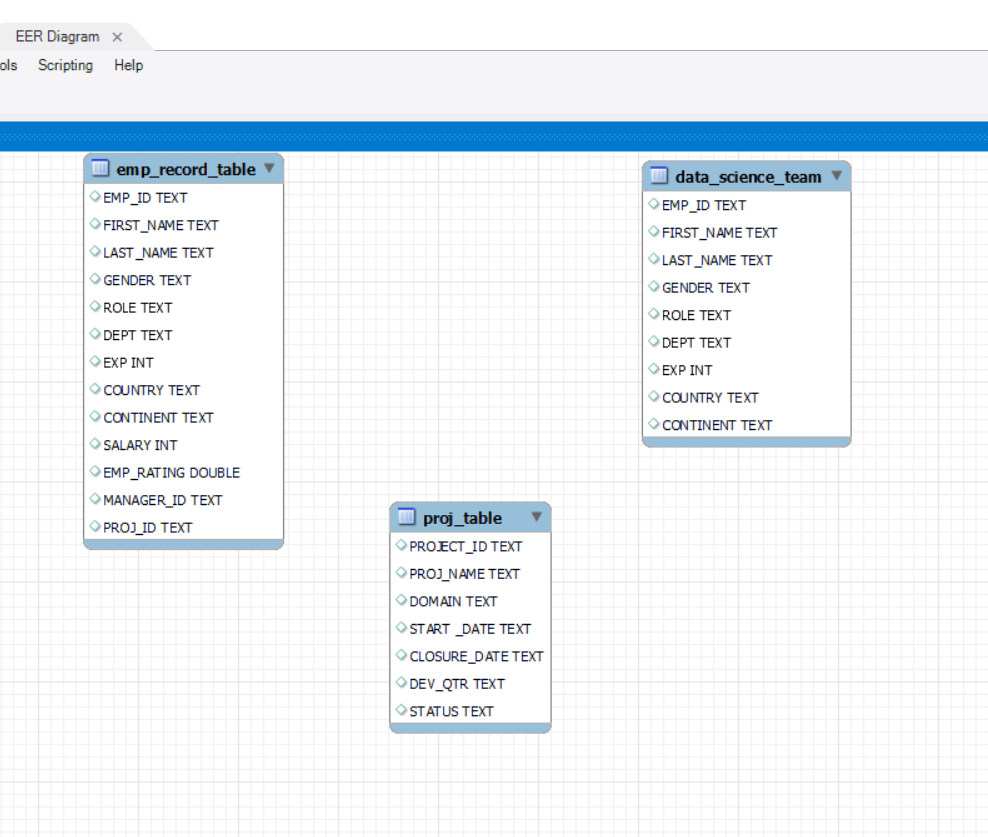
**SQL TRAINING PROJECT**

**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.

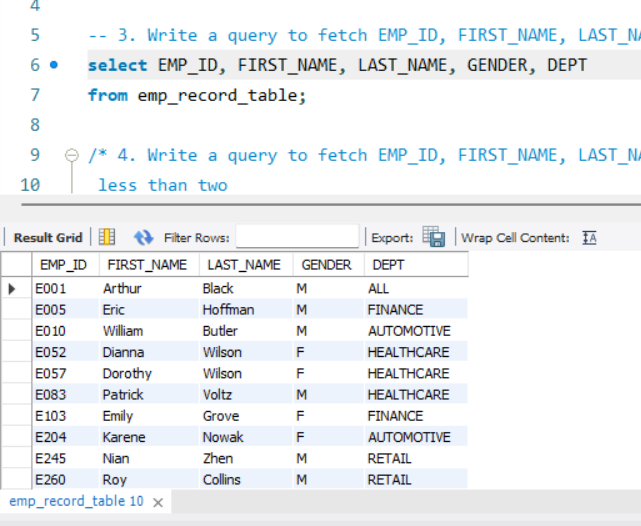
**QUERY:** create database employee;



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

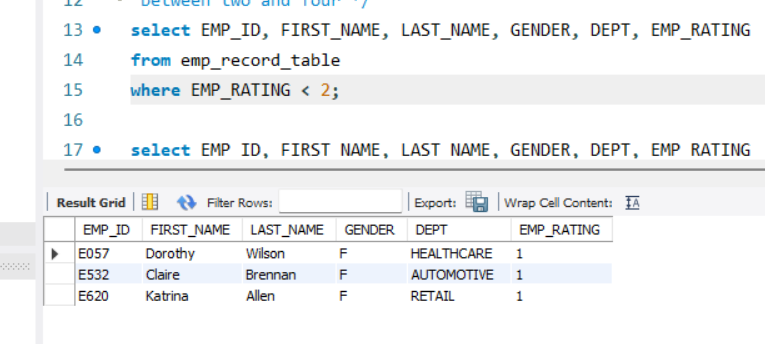
**QUERY:** select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT

from emp\_record\_table;



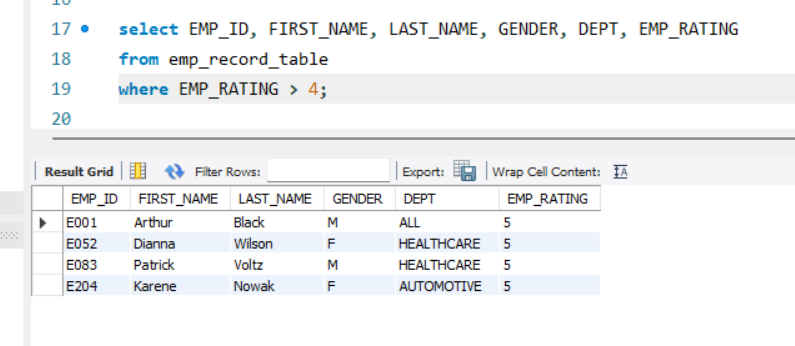
1. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:
   1. less than two
   2. greater than four
   3. between two and four
2. **QUERY:** select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING

from emp\_record\_table

where EMP\_RATING < 2;

1. **QUERY:** select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING

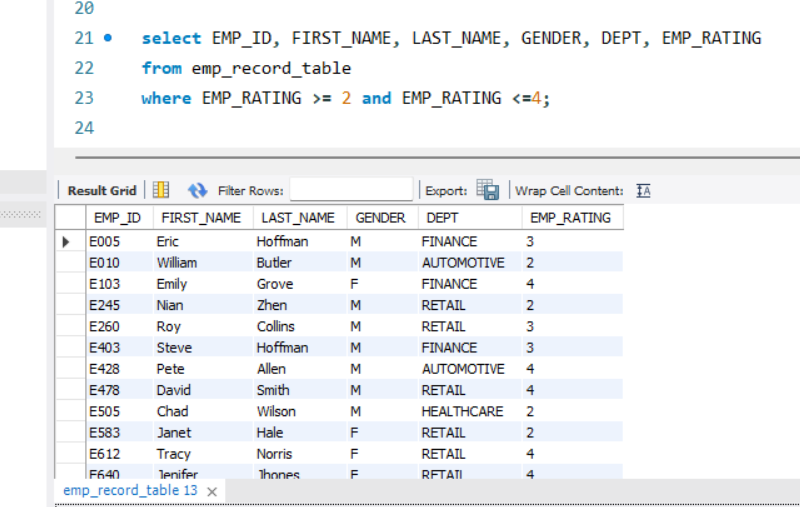
from emp\_record\_table

where EMP\_RATING > 4;

1. **QUERY:** select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING

from emp\_record\_table

where EMP\_RATING >= 2 and EMP\_RATING <=4;

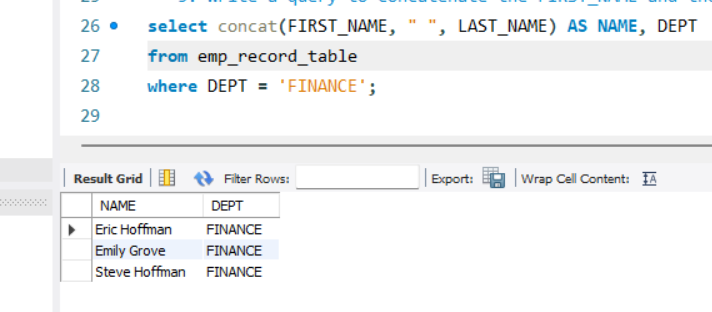


1. Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

**QUERY:** select concat(FIRST\_NAME, " ", LAST\_NAME) AS NAME, DEPT

from emp\_record\_table

where DEPT = 'FINANCE';



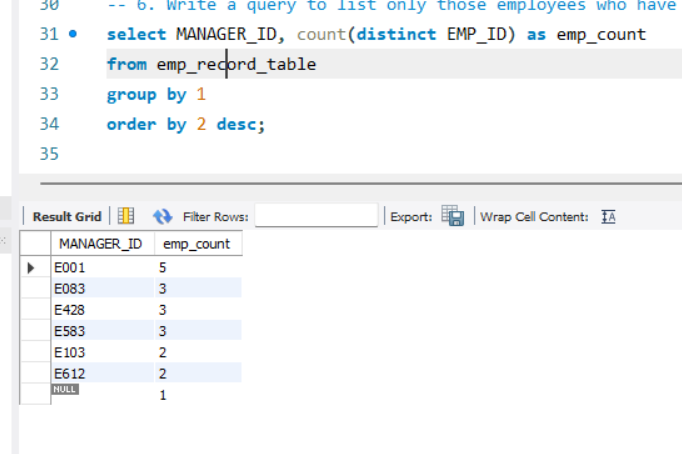
1. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).

**QUERY:** select MANAGER\_ID, count(distinct EMP\_ID) as emp\_count

from emp\_record\_table

group by 1

order by 2 desc;



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.

**QUERY:** select EMP\_ID, FIRST\_NAME, LAST\_NAME, DEPT

from emp\_record\_table

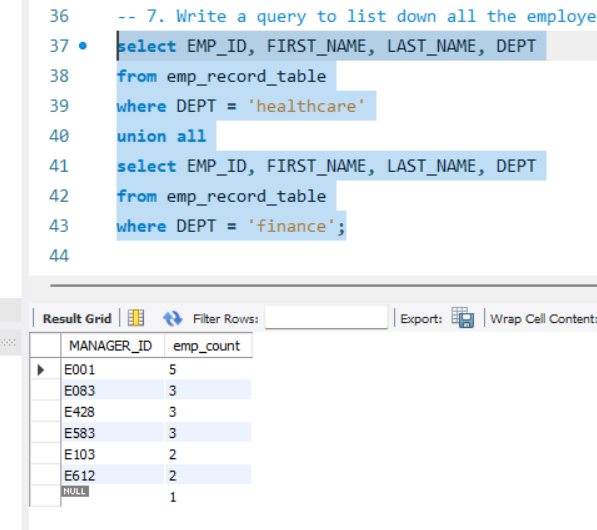
where DEPT = 'healthcare'

union all

select EMP\_ID, FIRST\_NAME, LAST\_NAME, DEPT

from emp\_record\_table

where DEPT = 'finance';



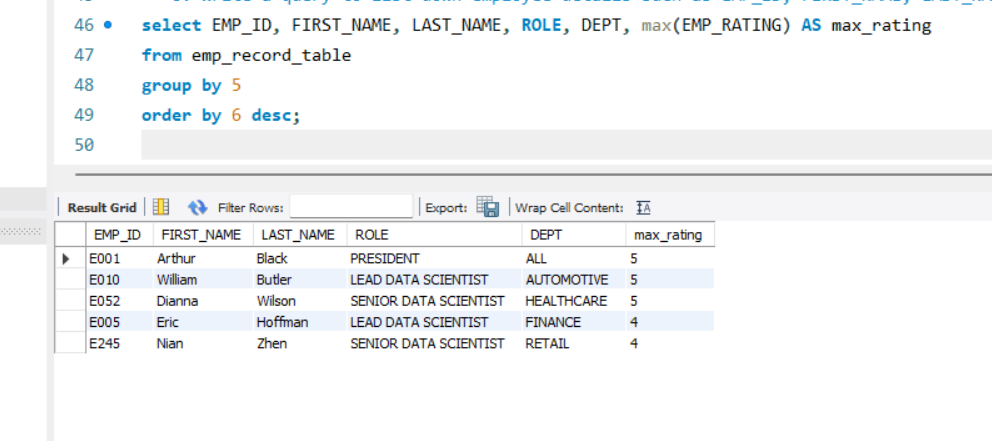
1. Write a query to list down employee details such as EMP\_ID, FIRST\_NME, LAST\_NAME, ROLE, DEPARTMENT, and EMP\_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department.

**QUERY:** select EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, max(EMP\_RATING) AS max\_rating

from emp\_record\_table

group by 5

order by 6 desc;

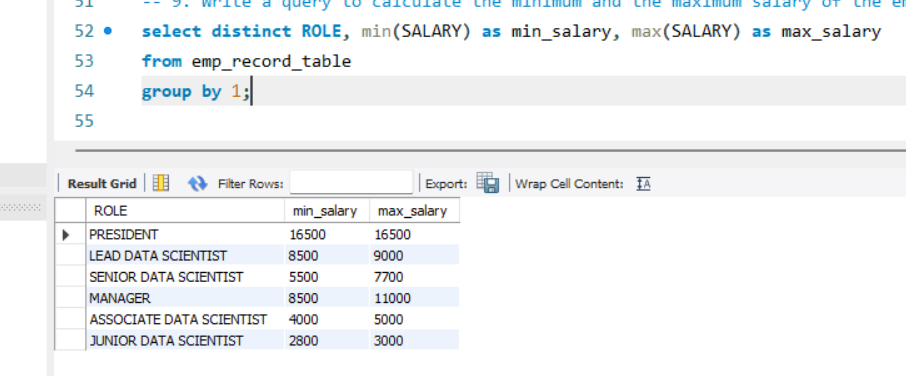


1. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

**QUERY:** select distinct ROLE, min(SALARY) as min\_salary, max(SALARY) as max\_salary

from emp\_record\_table

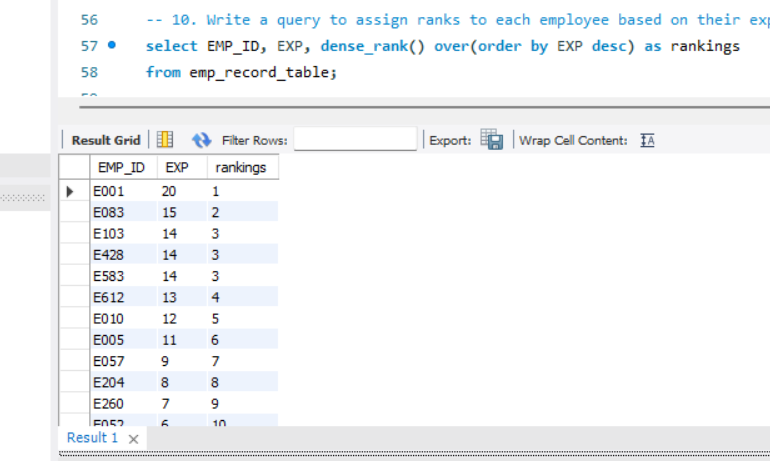
group by 1;



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

**QUERY:** select EMP\_ID, EXP, dense\_rank() over(order by EXP desc) as rankings

from emp\_record\_table;



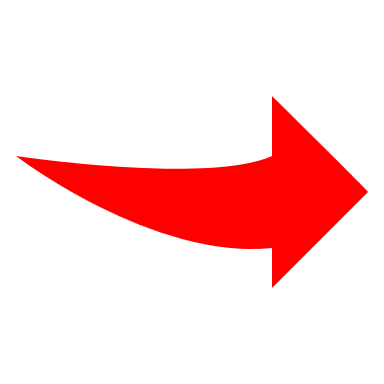
1. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.

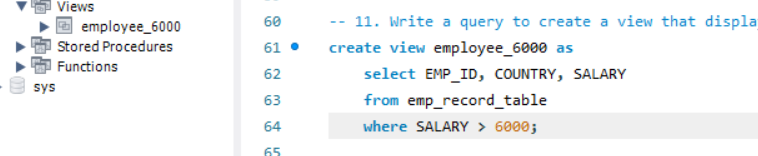
**QUERY:** create view employee\_6000 as

select EMP\_ID, COUNTRY, SALARY

from emp\_record\_table

where SALARY > 6000;

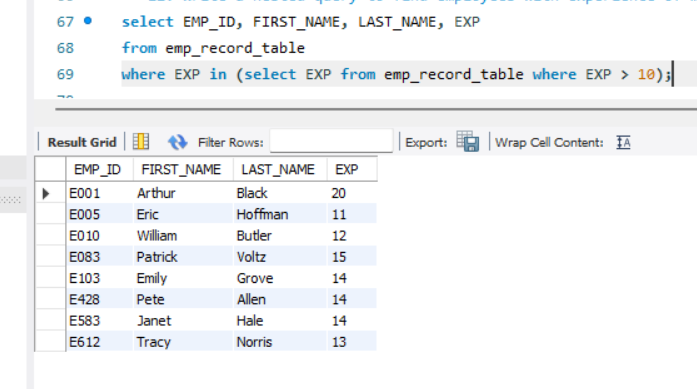




1. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

**QUERY:** select EMP\_ID, FIRST\_NAME, LAST\_NAME, EXP

from emp\_record\_table

where EXP in (select EXP from emp\_record\_table where 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.

**QUERY:** create procedure morethan\_3exp ()

begin

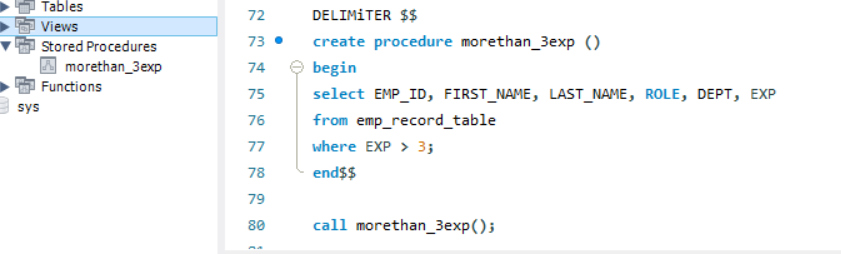
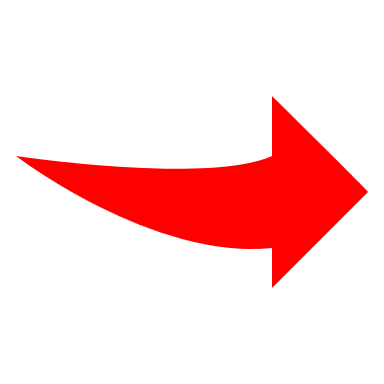
select EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, EXP

from emp\_record\_table

where EXP > 3;

end$$

call morethan\_3exp();



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'.

**QUERY:** DELIMITER $$

create function match\_profile (EXP int)

returns varchar(50)

deterministic

begin

declare ROLE varchar(50);

if EXP <= 2 then

set ROLE = 'JUNIOR DATA SCIENTIST';

elseif EXP between 2 and 5 then

set ROLE = 'ASSOCIATE DATA SCIENTIST';

elseif EXP between 5 and 10 then

set ROLE = 'SENIOR DATA SCIENTIST';

elseif EXP between 10 and 12 then

set ROLE = 'LEAD DATA SCIENTIST';

elseif EXP between 12 and 16 then

set ROLE = 'MANAGER';

end if;

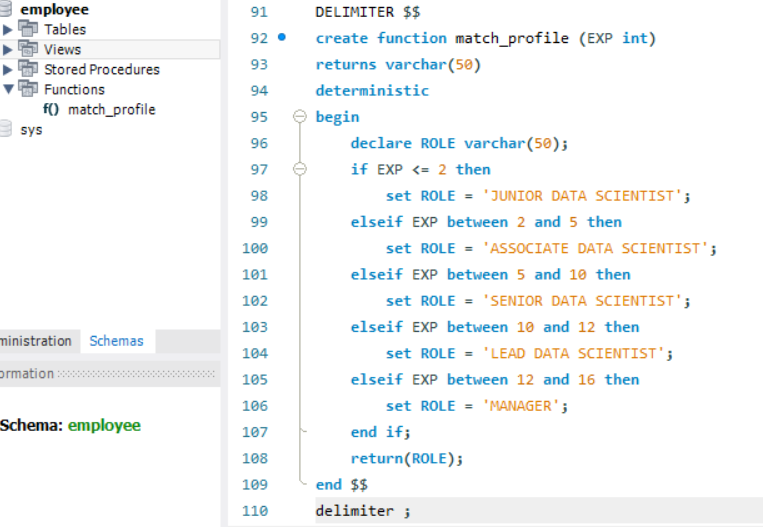
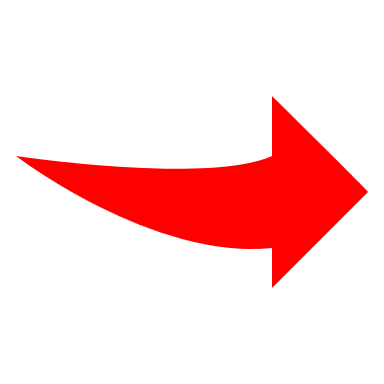
return(ROLE);

end $$

delimiter ;

select EMP\_ID, FIRST\_NAME, LAST\_NAME, EXP, match\_profile(EXP)

from emp\_record\_table;



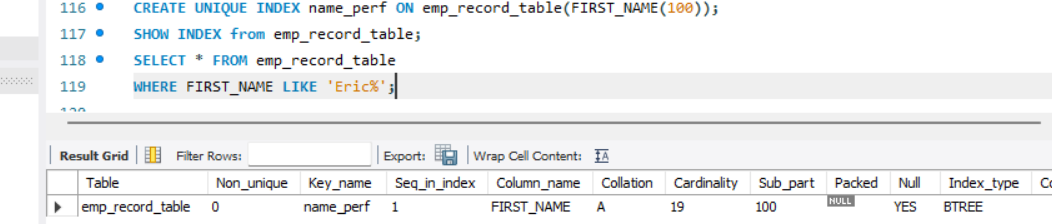
1. Create an index to improve the cost and performance of the query to find the employee whose FIRST\_NAME is ‘Eric’ in the employee table after checking the execution plan.

**QUERY:** CREATE UNIQUE INDEX name\_perf ON emp\_record\_table(FIRST\_NAME(100));

SHOW INDEX from emp\_record\_table;

SELECT \* FROM emp\_record\_table

WHERE FIRST\_NAME LIKE 'Eric%';

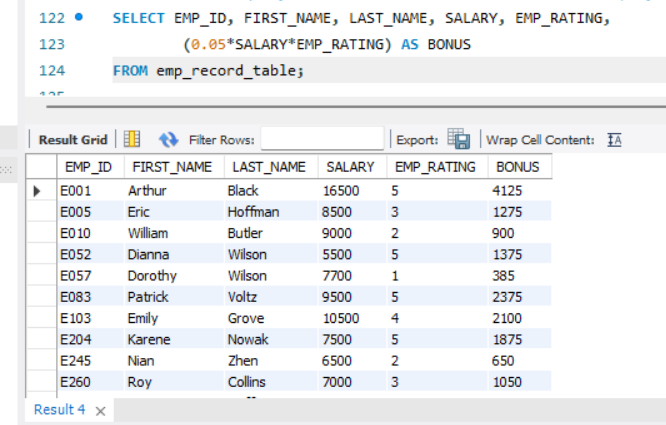


1. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).

**QUERY:** SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, SALARY, EMP\_RATING,

(0.05\*SALARY\*EMP\_RATING) AS BONUS

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.

**QUERY:** SELECT CONTINENT, COUNTRY, AVG(SALARY) AS AVERAGE\_SALARY

FROM emp\_record\_table

GROUP BY 1,2;

